**A Major Project Report on**

**SIGNATURE VERIFICATION USING MATLAB**

**Submitted to the Department of Computer Science & Engineering, GNITS in the partial fulfillment of the academic requirement for the award of B.Tech (CSE) under JNTUH**

By

G. Likitha Reddy 16251A05D4

G. Shreya 16251A05D9

L. Pravalika 16251A05E9

under the guidance of

Mrs.K.Sindhura

Asst. Professor

****

**Department of Computer Science & Engineering**

**G. Narayanamma Institute of Technology and Science**

**(Autonomous) (For Women)**

Shaikpet, Hyderabad- 500 104.

**Affiliated to**

**Jawaharlal Nehru Technological University Hyderabad**

Hyderabad – 500 085

Nov, 2019

**Department of Computer Science & Engineering**

**G. Narayanamma Institute of Technology and Science**

**(Autonomous) (For Women)**

**Accredited by NBA & NAAC &Affiliated to JNTUH**

**Hyderabad-500104, T.S, INDIA**

****

**CERTIFICATE**

This is to certify that the Project report on “**SIGNATURE VERIFICATION USING MATLAB**” is a bonafide work carried out by **G.Likitha - 16251A05D4, G.Shreya - 16251A05D9, L.Pravalika – 16251A05E9** in the partial fulfillment for the award of B.Tech degree in Computer Science & Engineering, G. Narayanamma Institute of Technology & Science, Shaikpet, Hyderabad, affiliated to Jawaharlal Nehru Technological University, Hyderabad under our guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

**Internal Guide Head of the Department**

**Mrs. K. Sindhura Dr. M. Seetha**

**Asst. Professor Professor & HOD**

**Department of CSE Department of CSE**

**External Examiner**

**ACKNOWLEDGEMENTS**

We would like to express our sincere thanks to **Dr.K.Ramesh Reddy, Principal** , GNITS, for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. M. Seetha, Professor and HOD**, Dept. of CSE, GNITS for all the timely support and valuable suggestions during the period of our major project.

We are extremely thankful to **Mrs.D.V.Lalitha Parameswari, Sr.Asst.Prof, Mrs. P. Neelima Reddy, Asst.Prof** Dept. of CSE, GNITS, the miniproject coordinators **and Dr. N. Kalyani, Professor,** Dept. of CSE, GNITS for their encouragement and support throughout the project.

We are extremely thankful and indebted to our internal guide, **Mrs.K.Sindhura, Asst.Prof,** Dept. of CSE, GNITSforher constant guidance, continuous advice, encouragement and moral support throughout the major project.

Finally, we would also like to thank all the faculty and staff of CSE Department who helped us directly or indirectly, parents and friends for their cooperation in completing the major project work.

G.Likitha - 16251A05D4

G.Shreya - 16251A05D9

L.Pravalika - 16251A05E9

# **ABSTRACT**

  Biometrics is currently perceived as a vital technology for setting up secure access control. Signature continue to be an important biometric for authenticating the identity of human beings. Handwritten signature is broadly utilized as personal verification in financial institutions. It is essential to determine the signature is original or fraud that ensures the necessity for a robust automatic signature verification tool. This tool aims to reduce fraud in all related financial transaction sectors.

The Image processing presents an effective method to perform handwritten signature verification using unique structural features extracted from the signature's contour. Once the image of a handwritten signature for a customer is captured, several pre-processing steps are performed on it including filtration and detection of the signature edges. Afterwards, a feature extraction process is applied on the image. Finally, a verification process is developed and applied to compare the extracted image features with those stored in the database for the specified customer.

The proposed tool is a robust and automatic signature verification system developed, using the recent advances in image processing and machine learning. Different machine learning techniques are applied to verify whether the signature is original or fraud. These techniques are also used to measure the euclidian distance in order to perform matching with original signature. The results of different techniques are compared with their accuracy and the best method is determined, to perform automatic signature verification.

Contents

[**ABSTRACT** iv](#_Toc24629041)

[**1.INTRODUCTION** 1](#_Toc24629042)

[**1.1 Objectives** 2](#_Toc24629043)

[**1.2 Methodology** 2](#_Toc24629044)

[**1.2.1 Algorithm Used** 3](#_Toc24629045)

[**1.3 Organization of the project** 4](#_Toc24629046)

[**2.THEORITICAL ANALYSIS OF THE PROPOSED PROJECT** 5](#_Toc24629047)

[**2.1 Existing Systems** 5](#_Toc24629048)

[**2.2 Proposed System** 5](#_Toc24629049)

[**2.3Advantages & Disadvantages** 6](#_Toc24629050)

[**2.3.1. Disadvantages of Existing System** 6](#_Toc24629051)

[**2.3.2. Advantages of Proposed System over Existing System** 6](#_Toc24629052)

[**2.4 Feasibility Analysis** 7](#_Toc24629053)

[**2.4.1. Feasibility study** 7](#_Toc24629054)

[**2.4.2. Economic Feasibility** 7](#_Toc24629055)

[**2.4.3. Technical Feasibility** 7](#_Toc24629056)

[**2.5 Requirement Analysis** 8](#_Toc24629057)

[**3.SYSTEM ARCHITECTURE** 9](#_Toc24629058)

[**3.1 Architecture** 9](#_Toc24629059)

[**3.2 Flow Diagram** 10](#_Toc24629060)

[**3.3 Module description** 11](#_Toc24629061)

[**4. IMPLEMENTATION AND CODING** 13](#_Toc24629062)

[**4.1 Implementation** 13](#_Toc24629063)

[**4.1.1 Code for GUI Screen** 13](#_Toc24629064)

[**4.1.2 Code for subject wise Result analysis histogram** 14](#_Toc24629065)

[**4.1.4 Code for re-evaluation results** 15](#_Toc24629066)

[**4.2. Description of technology used** 16](#_Toc24629067)

[**4.2.1. Python Qt Designer** 16](#_Toc24629068)

[**4.2.2. Languages (PYTHON)** 17](#_Toc24629069)

[**4.3. UML DIAGRAMS** 19](#_Toc24629070)

[**5. RESULTS** 24](#_Toc24629071)

[**5.1. System Testing Results** 24](#_Toc24629072)

[**5.2. Screenshots** 24](#_Toc24629073)

[**6. CONCLUSIONS AND FUTURE ENHANCEMENTS** 30](#_Toc24629074)

[**6.1 Conclusion** 30](#_Toc24629075)

[**6.2 Future Scope** 30](#_Toc24629076)

[**6.3 References** 31](#_Toc24629077)

# **INTRODUCTION**

As signature is the primary mechanism both for authentication and authorization in legal transactions, the need for efficient auto-mated solutions for signature verification has increased .Unlike a password, PIN, PKI or key cards – identification data that can be forgotten, lost, stolen or shared – the captured values of the handwritten signature are unique to an individual and virtually impossible to duplicate. Signature verification is natural and intuitive. The technology is easy to explain and trust. The primary advantage that signature verification systems have over other type’s technologies is that signatures are already accepted as the common method of identity verification.

A signature verification system and the techniques used to solve this problem can be divided into two classes Online and Off-line .On-line approach uses an electronic tablet and a stylus connected to a computer to extract information about a signature and takes dynamic information like pressure, velocity, speed of writing etc. for verification purpose. Offline signature verification involves less electronic control and uses signature images captured by scanner or camera. An offline signature verification system uses features extracted from scanned signature image. The features used for offline signature verification are much simpler. In this only the pixel image needs to be evaluated. But, the off-line systems are difficult to design as many desirable characteristics such as the order of strokes, the velocity and other dynamic information are not available in the off-line case. The verification process has to wholly rely on the features that can be extracted from the trace of the static signature images only.

Vigorous research has been pursued in handwriting analysis and pattern matching for a number of years. In the area of Handwritten Signature Verification (HSV), especially offline HSV, different technologies have been used and still the area is being explored. In this section we review some of the recent papers on offline HSV. The approaches used by different researchers differ in the type of features extracted, the training method, and the classification and verification model used.

## **Objectives**

1. As signature is the primary mechanism both for authentication and authorization in legal transaction.

2. The need for efficient auto-mated solutions for signature verification has increased.

3. The primary advantage that signature verification systems have over other type’s technologies is that signatures are already accepted as the common method of identity verification.

## **Methodology**

In this section, block diagram of system is discussed. Fig. 1 gives the block diagram of proposed signature verification system which verifies the authenticity of given signature of a person. The design of a system is divided into two stages:

1) Training stage

2) Testing stage

A training stage consist of four major steps

1) Retrieval of a signature image from a database

2) Image pre-processing

3) Feature extraction

4) Neural network training

A testing stage consists of five major steps

1) Retrieval of a signature to be tested from a database

2) Image pre-processing

3) Feature extraction

4) Application of extracted features to a trained neural network 5) Checking output generated from a neural network.

### 

### **Fig 1**

### **1.2.1 Algorithm Used**

**Algorithm for Handwritten Signature Verification using Neural Network**

Input: signature from a database

Output: verified signature classified as genuine or forged.

1. Retrieval of signature image from a database.

2. Preprocessing the signatures.

2.1 Converting image to binary.

2.2 Image resizing.

2.3 Thinning.

2.4 Finding bounding box of the signature.

3. Feature extraction

3.1 Maximum horizontal and vertical histogram

3.2 Centre of mass

3.3 Normalized area of signature

3.4 Aspect ratio

3.5 The tri surface feature

3.6 The six fold surface feature

3.7 Transition feature

4. Creation of feature vector by combining extracted features.

5. Normalizing a feature vector.

6. Training a neural network with a normalized feature vector.

7. Steps 1 to 5 are repeated for testing signatures.

8. Applying normalized feature vector of test signature to trained neural network.

9. Using a result generated by the output neuron of the neural network declaring a signature as a genuine or forged.

## **1.3 Organization of the project**

The organization has 6 chapters which gives a brief description about the all the modules in the system, namely Faculty Details, Class Details, Result analysis, Re-evaluation.

In the faculty details module, we have the information about the faculty of the particular section. All the names of the faculty are shown in this module. Faculty must also be authorized users of the college. Faculty can view the class result, subject wise result. On clicking this module, the faculty names will be shown. When we click on particular faculty the corresponding faculty subject graph will be displayed which shows the histogram on a scale of 4-10 that indicates the grade points of the students in subject taught by the faculty. The percentages of students pass and fail in a subject can be viewed.

Every Department consist of 4 years of students with three sections of each year. To analyze each section details we send an excel sheet as the input. On clicking the class details button the screen will be displayed to select corresponding section upon selecting section the details of class we be sent into new excel sheet containing the number of students with percentage greater than 90, greater than 80…in each subject taught in the semester.

In the result analysis, a report is generated (an excel sheet), which makes an easy understanding of the performance of the students. It displays a graph of each subject corresponding to grades. This makes us realize on pass percentage of each subject, subject with highest failures, and the subject with good percentages

The students who are not satisfied with the result they obtained have a chance to apply for re-evaluation. The students applied for the re-evaluation, will receive their result after some days in a separate excel sheet. These students’ results will be compared with the original excel sheet and new excel sheet report of modified result will be generated.

# **2.THEORITICAL ANALYSIS OF THE PROPOSED PROJECT**

## 

## **2.1 Existing Systems**

**Existing Method:-**

**Features Extraction process**

* Static approaches-The static one involves geometric measures of the signature,
* Pseudo-dynamic approaches-Automatic static handwritten signature verification based on the use of gray level values from signature stroke pixels.

## **2.2 Proposed System**

**Proposed Method:-**

Behavioral Biometric Application

* Online signature (Data Set Images)
* Template aging,
* Performance evaluation,

Verification

Similarity Measurement

Features Extraction

DATA Set Images

Features Extraction

Segmentation

Pre-processing

Input Test Image

## **2.3Advantages & Disadvantages**

### **2.3.1. Disadvantages of Existing System**

* Accuracy is less.
* Illumination occurrences is less.
* Error Rate is more.

### **2.3.2. Advantages of Proposed System over Existing System**

* User friendly (as faculties can easily use application).
* Easy computation.
* Provide accurate results.
* Reduce manual work.
* Requires less effort and time.
* The software is time and space efficient.

## **2.4 Feasibility Analysis**

# **2.4.1.** **Feasibility study**

As the name implies, a feasibility study is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn by taking on a model that is not profitable.

* Ability to track & analyse the subject average by various combinations like Exam - wise & class - wise.
* Statistical reports like Student Mark Comparison & Student list can be taken under various criteria as per our requirement.

### **2.4.2. Economic Feasibility**

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision makers determine the positive economic benefits to the organization that the proposed project will provide. Our project is economically feasible because in this we have used “WINDOWS”, “PYTHON”, “PYQT” designer tool and “PYUIC” which are all available as an open source.

### **2.4.3. Technical Feasibility**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system.

### **2.5 Requirement Analysis**

**Software Requirements:**

It requires a 64-bit Windows Operating System.

Python Qt Designer for designing user interface.

PyUIC for converting the layout designed user interface (UI) to python code.

Python language for coding.

**Hardware Requirements:**

It requires a minimum of 2.16 GHz processor.

It requires a minimum of 4 GB RAM.

It requires 64-bit architecture.

It requires a minimum storage of 500GB.

# **3.SYSTEM ARCHITECTURE**

# **3.1 Architecture**

# https://www.researchgate.net/profile/Vahab_Iranmanesh/publication/264312002/figure/fig1/AS:213800780341274@1427985486835/Online-signature-verification-system-schema.png

Fig: 3.1 Architecture of Signature verification system.

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

[,].

In the last decade, a number of studies have been carried

out on online and oine signature verication with the

sole aim of improving the verication accuracy []. e

verication system should also incorporate lesser compu-

tational complexity in order to provide fast response for

real-time applications []. Several classication methods

have been suggested for robust verication purposes, one

of which includes articial neural network (ANN) [–

]. us, in this paper, we maintain the use of ANN

as the classier and focus on improvement at the feature

level

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system []. Initially, signature

samples are collected in the enrollment stage, whereby

useful information known as dynamic features is extracted

in order to build a user’s reference template, which is

stored in the knowledge database. en, the template

is used as a reference for comparison with the new

queried user’s features to decide either to reject or to

accept the queried signature sample as genuine or not

[]. It is virtually impossible for a user to reproduce his/her

exact signature on multiple attempts due to intrauser variabil-

ity. Intrauser variability measures the dierence between the

signatures of an individual, which may be inuenced by envi-

ronmental, health, and emotional challenges while signing

Figure  depicts the basic structural design of an online

signature verication (OSV) system

Figure  depicts the basic structural design of an online

signature verication (OSV) system

Figure  depicts the basic structural design of an online

signature verication (OSV) system

Figure  depicts the basic structural design of an online

signature verication (OSV) system

Fig 3.1 depicts the basic structural design of an online verification system[OVS].Initially signature samples are collected in enrollment stage, whereby useful features such as dynamic features are extracted in order to build a users reference template,Which is stored in knowledge database. Then the emplate is

## **3.2 Flow Diagram**

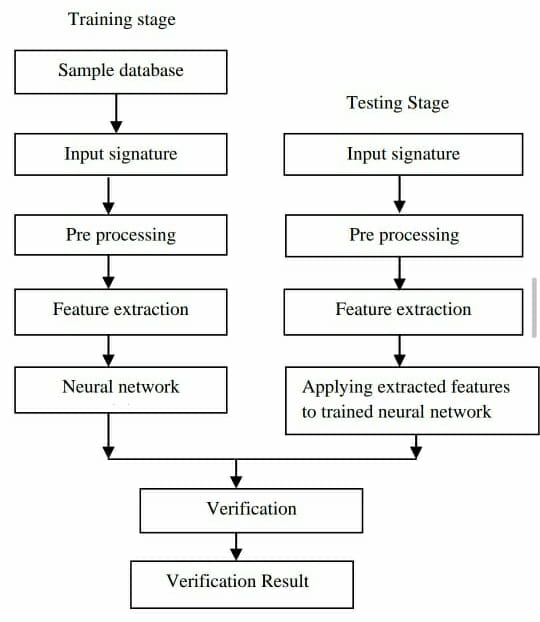
****

Fig: 3.2 Flow Diagram

The above flow chart fig 3.2 describes the flow of the process of the system. It begins with user followed by the Interface. After which system offers various choices to analyze the results. The class details provide the particular section grades. The faculty details display the graph of the subject taught by the faculty. The subject details show the subject wise failures, subject wise grades and subject wise percentages. The re-evaluation display the updated excel after comparing two excel sheets.

## **3.3 Module description**

There are 4 modules in this application

* Faculty Details
* Class Details
* Result analysis
* Re-evaluation

**Faculty Details:**

In this module, we have the information about the faculty of the particular section.

All the names of the faculty are shown in this module.

Faculty must also be authorized users of the college. Faculty can view the class result, subject wise result. On clicking this module, the faculty names will be shown. When we click on particular faculty the corresponding faculty subject graph will be displayed which shows the histogram on a scale of 4-10 that indicates the grade points of the students in subject taught by the faculty. The percentages of students pass and fail in a subject can be viewed.

**Class Details:**

Every Department consist of 4 years of students with three sections of each year. To analyze each section details we send an excel sheet as the input. On clicking the class details button the screen will be displayed to select corresponding section upon selecting section the details of class we be sent into new excel sheet containing the number of students with percentage greater than 90 , greater than 80…in each subject taught in the semester.

**Result analysis:**

In this, a report is generated (a excel sheet), which makes an easy understanding of the performance of the students. It displays a graph of each subject corresponding to grades. This makes us realize on pass percentage of each subject, subject with highest failures, and the subject with good percentages.

**Re-evaluation:**

The students who are not satisfied with the result they obtained have a chance to apply for re-evaluation. The students applied for the re-evaluation, will receive their result after some days in a separate excel sheet. These students’ results will be compared with the original excel sheet and new excel sheet report of modified result will be generated.

# **4. IMPLEMENTATION AND CODING**

## **4.1 Implementation**

### **4.1.1 Code for GUI Screen**

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui\_MainWindow(object):

def setupUi(self, MainWindow):

MainWindow.setObjectName("MainWindow")

MainWindow.resize(832, 676)

MainWindow.setStyleSheet("#centralwidget{\n"

" background-color: qlineargradient(spread:pad, x1:0.295455, y1:0.335, x2:1, y2:1, stop:0 rgba(255, 170, 170, 255), stop:1 rgba(127, 223, 255, 255));\n"

"}")

self.centralwidget = QtWidgets.QWidget(MainWindow)

self.centralwidget.setObjectName("centralwidget")

self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_2.setGeometry(QtCore.QRect(280, 260, 261, 51))

self.pushButton\_2.setStyleSheet("font: 10pt \"MS Shell Dlg 2\";")

self.pushButton\_2.setObjectName("pushButton\_2")

self.pushButton\_3 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_3.setGeometry(QtCore.QRect(280, 360, 261, 51))

self.pushButton\_3.setStyleSheet("font: 10pt \"MS Shell Dlg 2\";")

self.pushButton\_3.setObjectName("pushButton\_3")

self.pushButton\_4 = QtWidgets.QPushButton(self.centralwidget)

self.pushButton\_4.setGeometry(QtCore.QRect(280, 460, 261, 51))

self.pushButton\_4.setStyleSheet("font: 10pt \"MS Shell Dlg 2\";")

self.pushButton\_4.setObjectName("pushButton\_4")

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setGeometry(QtCore.QRect(320, 150, 341, 51))

self.label.setStyleSheet("font: 75 14pt \"Arial\";\n"

"text-decoration: underline;\n"

"font: 75 16pt \"MS Shell Dlg 2\";\n"

"text-decoration: underline;")

self.label.setObjectName("label")

MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(MainWindow)

QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):

\_translate = QtCore.QCoreApplication.translate

MainWindow.setWindowTitle(\_translate("MainWindow", "Results Analysis"))

self.pushButton\_2.setText(\_translate("MainWindow", "Subject wise Failures"))

self.pushButton\_3.setText(\_translate("MainWindow", "Section Wise Percentages"))

self.pushButton\_4.setText(\_translate("MainWindow", "subject details"))

self.label.setText(\_translate("MainWindow", "STATISTICS"))

### **4.1.2 Code for subject wise Result analysis histogram**

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

res1\_file = 'results2.xls'

results = pd.read\_excel(res1\_file)

data = results['CP'].tolist()

bins = [0,4,5,6,7,8,9,10]

hist, bin\_edges = np.histogram(data,bins) # make the histogram

fig,ax = plt.subplots()

# Plot the histogram heights against integers on the x axis

ax.bar(range(len(hist)),hist,width=1)

# Set the ticks to the middle of the bars

ax.set\_xticks([0.5+i for i,j in enumerate(hist)])

# Set the xticklabels to a string that tells us what the bin edges were

ax.set\_xticklabels(['{} - {}'.format(bins[i],bins[i+1]) for i,j in enumerate(hist)])

plt.show();

### **4.1.4 Code for re-evaluation results**

import openpyxl

def insert\_into\_dataset(predictions):

wb = openpyxl.load\_workbook('reval1.xlsx')

sheet1=wb.active

max\_row = sheet1.max\_row # Get max row of first sheet

max\_col = sheet1.max\_column # Get max column of first sheet

row=max\_row+1

for col\_num in range(1, max\_col + 1):

cell1 = sheet1.cell(row=row, column=col\_num)

cell1.value = predictions[col\_num-1]

wb.save('reval1.xlsx')

wkb1 = openpyxl.load\_workbook("sample1.xlsx")

wkb2 = openpyxl.load\_workbook("sample2.xlsx")

ws1 = wkb1.active

wkb1\_LastRow = ws1.max\_row

ws2 = wkb2.active

wkb2\_LastRow = ws2.max\_row

for xrow in range (1,(wkb1\_LastRow+1)):

for yrow in range (1,(wkb2\_LastRow+1)):

if ws1.cell(row=xrow,column=1).value == ws2.cell(row=yrow,column=3).value and ws1.cell(row=xrow,column=3).value == ws2.cell(row=yrow,column=4).value:

if ws1.cell(row=xrow,column=4).value != ws2.cell(row=yrow,column=6).value:

#print(ws1.cell(row=xrow,column=1).value,ws1.cell(row=xrow,column=3).value,ws1.cell(row=xrow,column=4).value, ws2.cell(row=yrow,column=6).value)

l=[]

l.extend([ws1.cell(row=xrow,column=1).value,ws1.cell(row=xrow,column=3).value,ws1.cell(row=xrow,column=4).value, ws2.cell(row=yrow,column=6).value])

insert\_into\_dataset(l)

## **4.2. Description of technology used**

This Project uses PyQt tool to create the needed Graphical User Interfaces, PyUIC module to automatically generate the automated code and python image Processing library(Pillow). PyQt is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt is developed by the British firm Riverbank Computing. PyQt supports Microsoft Windows as well as various flavors of UNIX, including Linux and MacOS. PyQt implements different classes and methods including classes for accessing SQL databases (ODBC, MySQL, Oracle, SQLite) PyUIC tool is used to automatically generate the code for the Front end user interfaces created by PyQt. All the front end python code is automatically generated by this tool, by converting the user interface(.ui) files into .py files.

### **4.2.1. Python Qt DesignerC:\Users\Dell PC\Desktop\qt.jpg**

The PyQt installer comes with a GUI builder tool called Qt Designer. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.

Creation of a GUI interface using Qt Designer starts with choosing a top-level window for the application. You can then drag and drop required widgets from the widget box on the left pane. You can also assign value to properties of widget laid on the form.

The designed form is saved as demo.ui. This ui file contains XML representation of widgets and their properties in the design. This design is translated into Python equivalent by using pyuic4 command line utility. This utility is a wrapper for ui module.

### **4.2.2. Languages (PYTHON)**

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing.

Python is an interpreted high-level programming language for general-purpose programming. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. Python is an open source software and has a community-based development model, as do nearly all of its variant implementations. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

**Steps for installing required packages in Python :**

1) Install Python by down loading it from the following site.

**https://www.python.org/downloads/release/python-370/**

Download Windows x86-64 executable installer...if your laptop is a 64-bit machine.

During installation, Check the radio box(with a tick mark) corresponding to Add path...

Note: Anaconda's python is not enough. You should have a separate Python.

2) After successful installation of python, open the command prompt(CMD), as administrator, and give the command **python -V**. If python version is displayed, then your installation is successful. If your installation is not successful, then skip the following 3,4,5,6 commands.

3) Execute the command from command prompt(CMD): **pip3 install wheel** you need to type 'y', if the system asks you to choose y/n. If the above command don't work, then use: **pip install wheel**

4) Execute the command from command prompt(CMD): **pip3 install pyqt5**. If the above command don't work, then use: **pip install pyqt5**

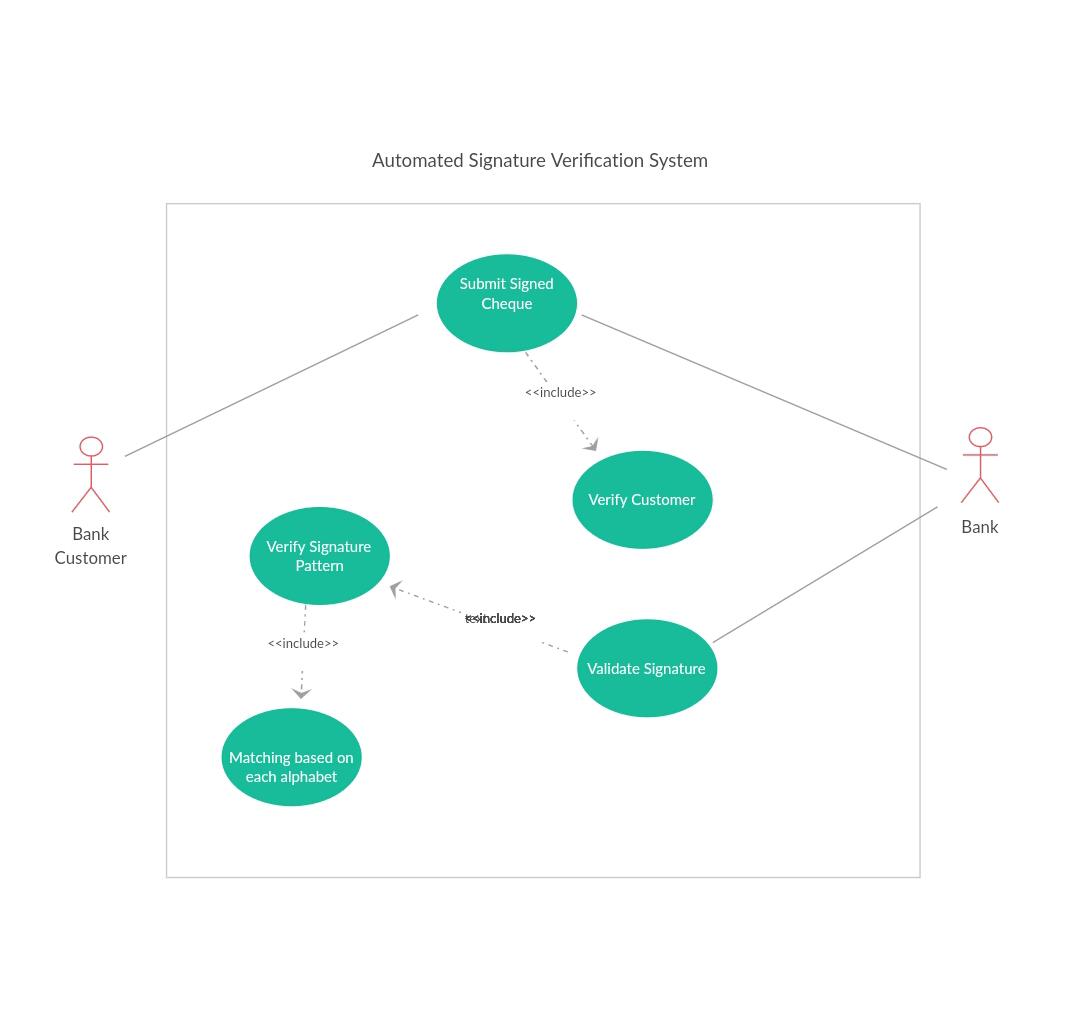
5) Execute the command from command prompt(CMD): **pip3 install pillow**. If the above command don't work, then use: **pip install pillow**

6) Execute the command from command prompt(CMD): **pip3 install opencv-python**. If the above command don't work, then use: **pip install opencv-python**

7) Installing Anaconda : Install Anaconda by downloading it from: [**https://www.anaconda.com/download/**](https://www.anaconda.com/download/)

## 

## **4.3. UML DIAGRAMS**



**Fig:** 4.1 Use Case Diagram for Signature verification System.

The fig 4.1 comprise of two actors namely admin, the faculty, and various use cases. The admin in this system refers to the user for adding faculty details and class details. Faculty is the actor that refers to teachers and professors to analyze the result.

The login usecase allows user to use the model by running it and start using it. Add class details allows you to add each section student and subject details in the model. Add Faculty details allows user to add subjects taught by the faculty. Result Analysis produces result of graph of each subject.

Reval is the usecase that compares two excel sheets and updates result in a new excel sheet. Chosen section allows user to choose a section to display section percentages of students in each subject. View section wise failures usecase allows viewing all the failures section wise of the students. View section percentages allows viewing all the percentages section wise of the students. View section wise details allows viewing all the details section wise of the students.

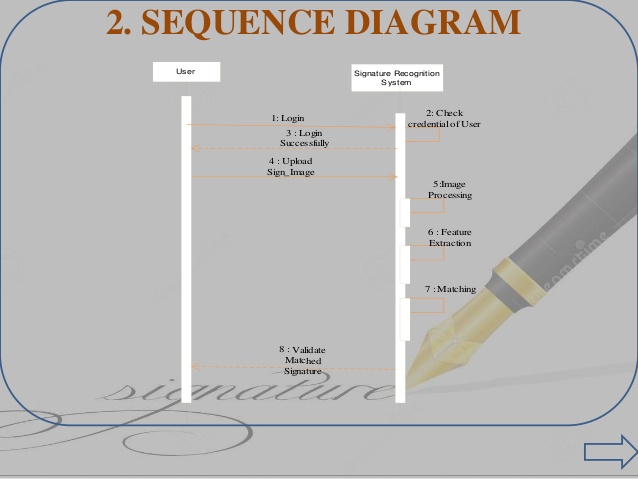
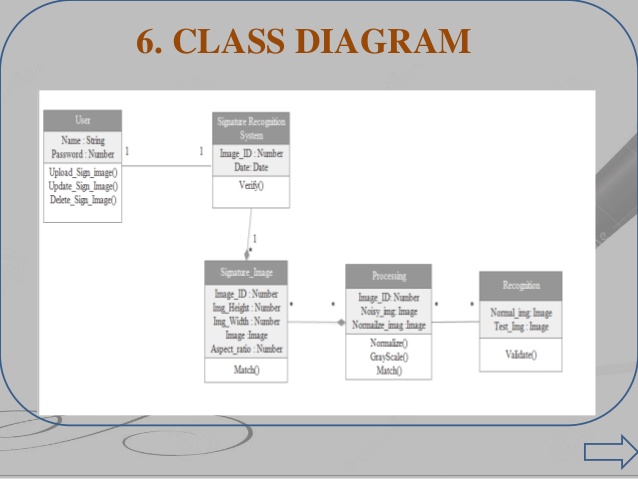


Fig: 4.2 Sequence Diagram for Result Analysis System

The above fig 4.2 is the sequence diagram comprises of actions performed by the user and system. Run application action is related to an action to start the application and begin to use it. Display options action specifies the option present in the application and allows choosing any required options. Click class Details is an action to check the details of the class.

Show details is an action from the system to display the details asked by the user. Select faculty details is an action where user selects any of the faculty to check faculty details. Displays graph is an action application displays a graph of faculty. Select subject Details action to check graphs of each subject and look at subject wise analysis.

Show result grades is an action the application displays excel sheet of grades. Check re-evaluation is an action the application compares results, which are in form of, excels and updates. Displays excel results is an action updated excel sheet is displayed.



**Fig:** 4.3 Class diagram for Result Analysis System

The above fig 4.3 is the class diagram that comprises various classes methods and attributes. System is the class that contains methods to analyze the result. The attributes in the class system are faculty details, subject details, reval and sections. Methods of system are runapplication () and displayoptions ().

Class details refers to details of students in class. The attributes of class details are sections and grades. The methods of class details are showgrades and showgraph. Faculty details is the class refers to details of faculty in class. The attributes of faculty details are names and grades. The methods of faculty details are showgraph.

Subject details is the class refers to details of each subject taught in class. The attributes of subject details are subjectfailures, subject percentages and subject grades. The methods of subject details are showsubjectfailures, showsujectpercentages, and showgrades.

Re-Evaluation is theclass refers to the reevaluation details of all students. The attributes of Re-evaluation are excel. The methods of Re-evaluation are compareresult, displayexcel.

# **5. RESULTS**

## **5.1. System Testing Results**

The project is thoroughly tested by testing each text box and push buttons of the GUI Screens and verifying the corresponding results in excel sheet.

The results are obtained in various formats some are in form of histograms, few results are obtained in new excel sheet which makes it easy to compare and understand and to take printouts of results.

## **5.2. Screenshots**

Following screen describes main GUI for the application

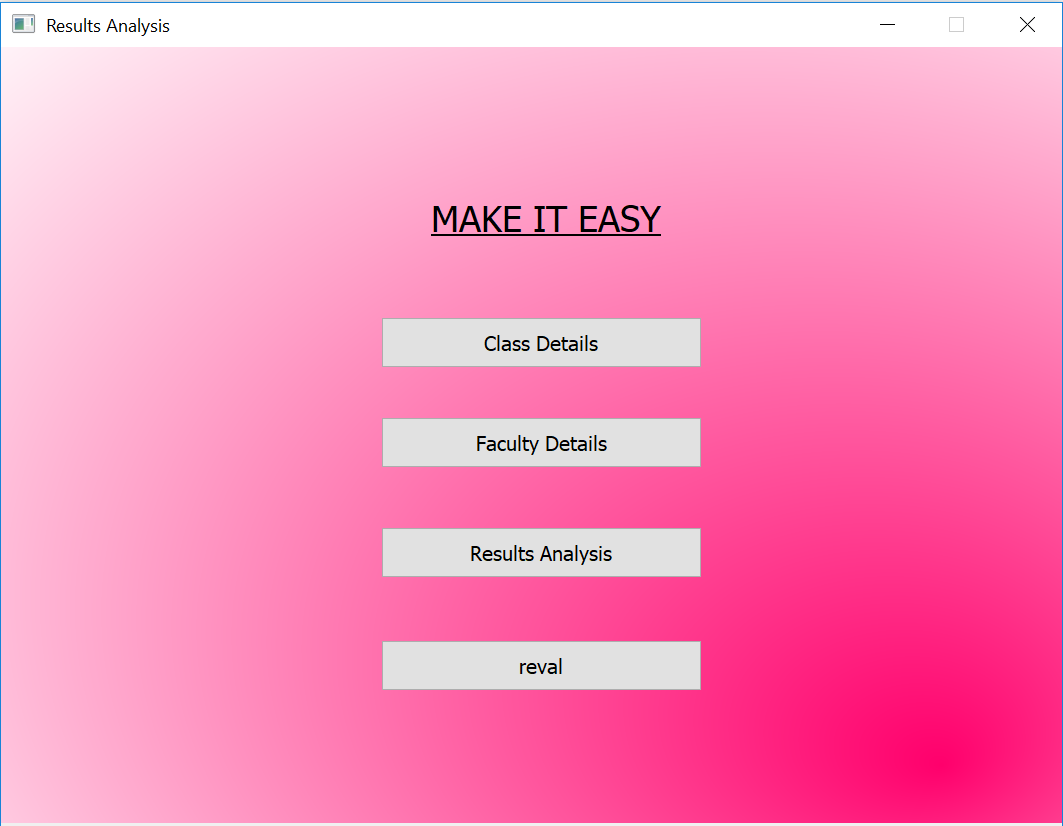


Fig 5.1 Main Screen

The above Fig 5.1 illustrates the first screen of the project. It contains title of the project. It contains four buttons to navigate to Class Details, Faculty Details, and Result Analysis, reval to analyze the result.

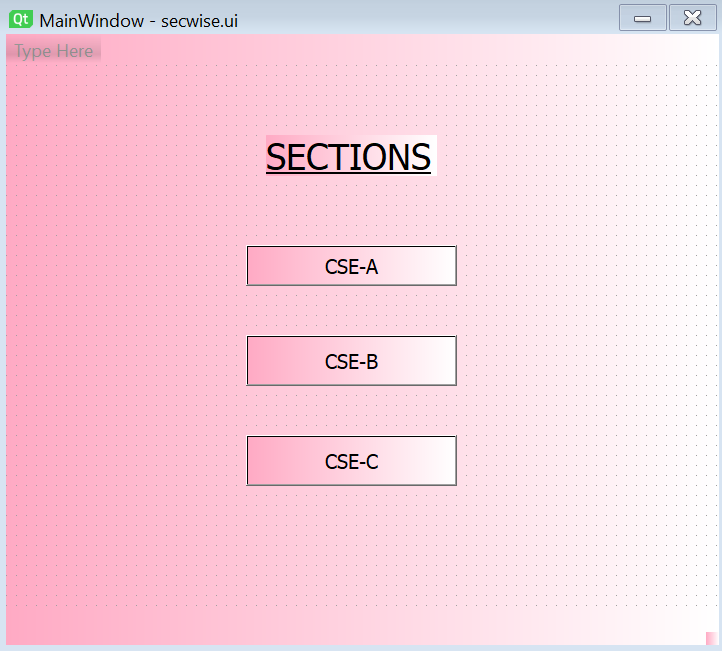


Fig 5.2 Choose Section

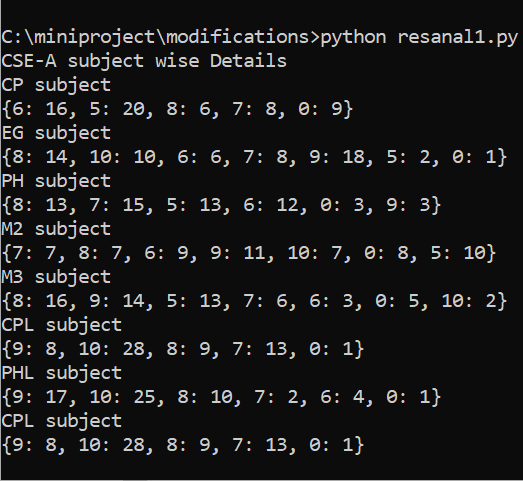


Fig 5.3 Section Details

The fig 5.2 displayed on choosing class details button. The system provides an option to select section out of the three sections mentioned in the tab.

After choosing the section, it displays section wise results as shown in fig 5.3.

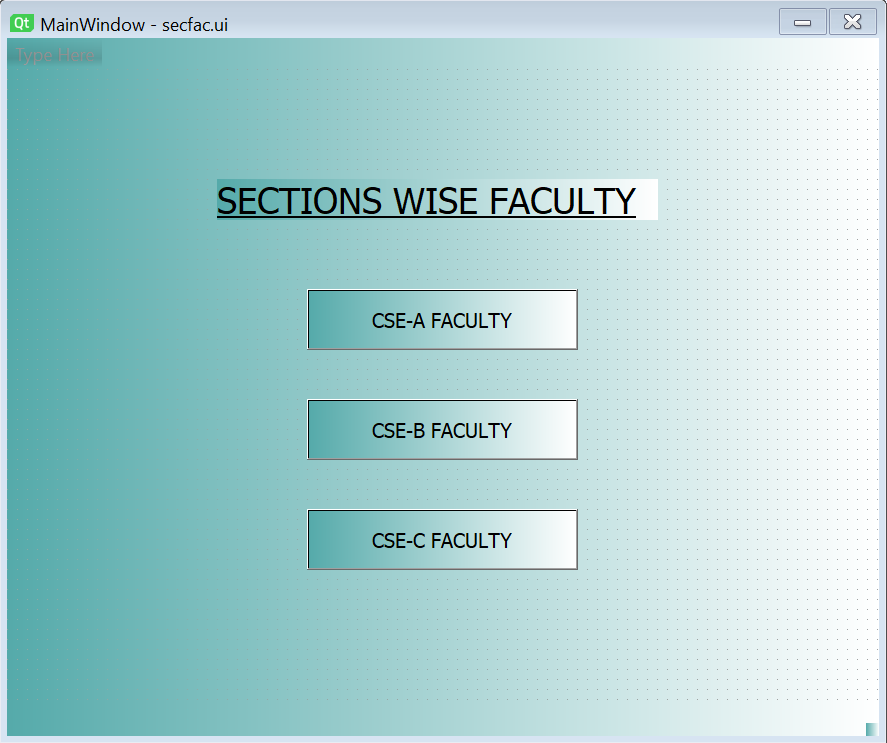


Fig 5.4 Choose Faculty

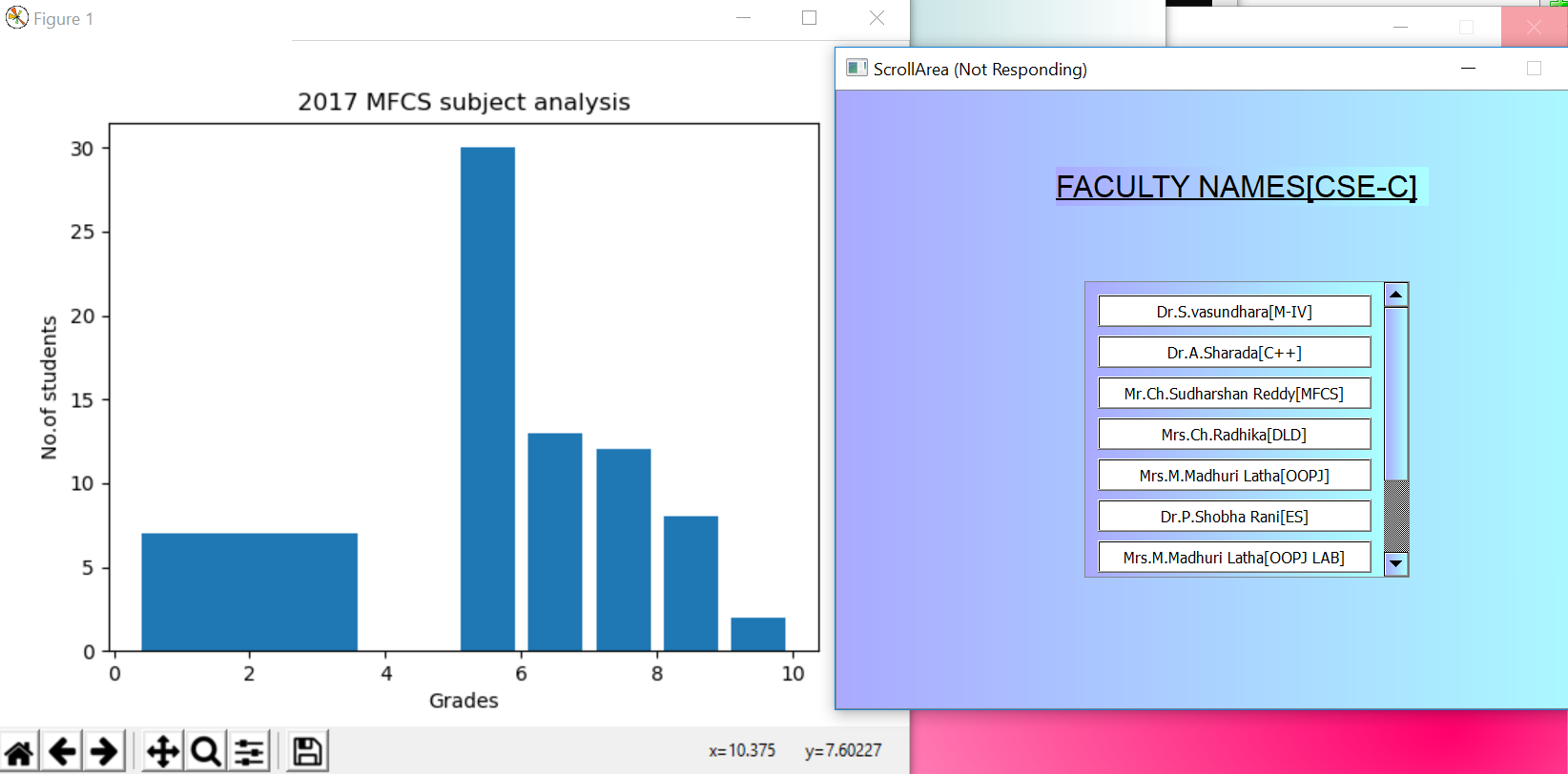


Fig 5.5 Faculty Results

When Faculty details is selected then the system displays fig 5.4 to choose any faculty from the three sections mentioned there. After choosing any particular faculty member, a graph is displayed about the faculty and the subject taught graphical Analysis.

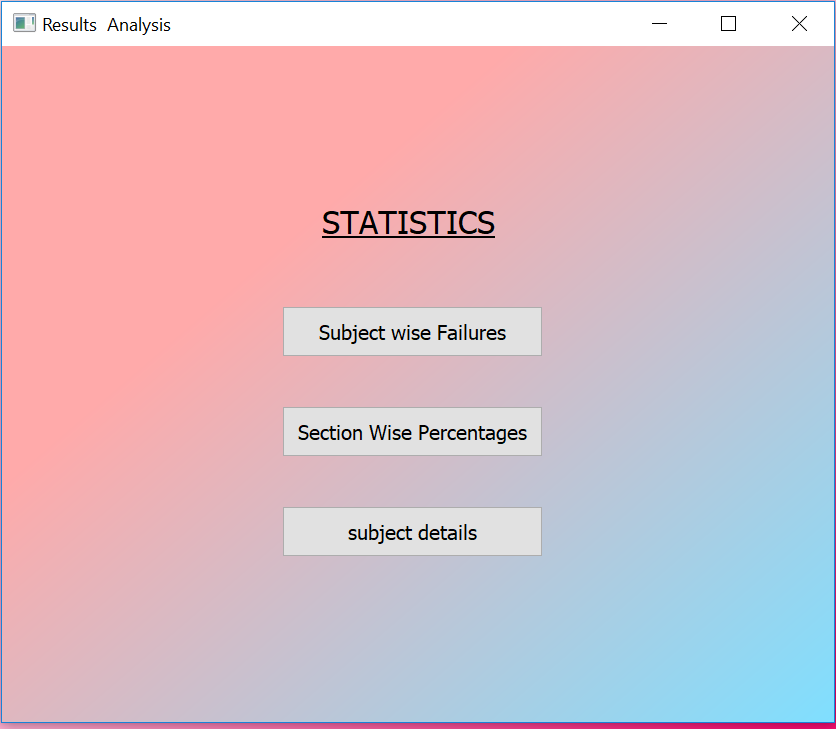


Fig 5.6: Subject Details

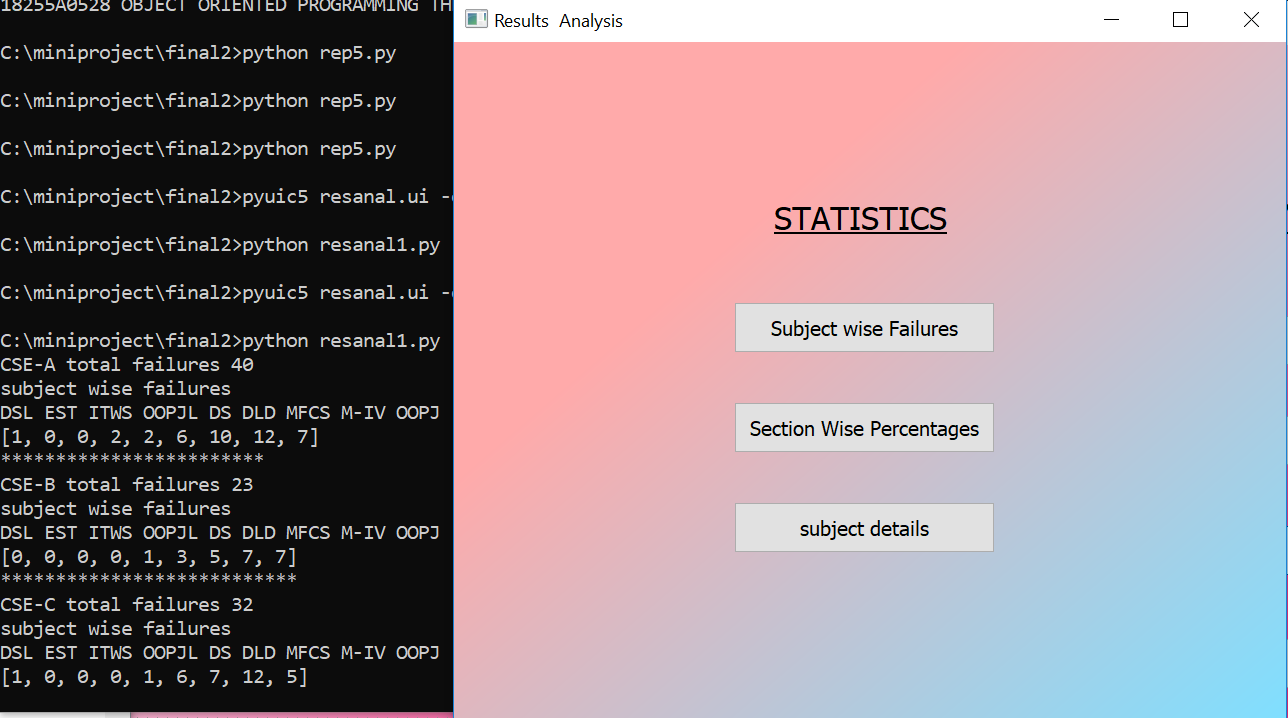


Fig 5.7: Subject wise Failures

Fig 5.6 is displayed when Result Analysis is selected. Result analysis button illustrates three features those are subject wise failures, section wise percentages, and subject details. One of the mentioned buttons are selected. Fig 5.7 is displayed if the subject wise failures button is selected.

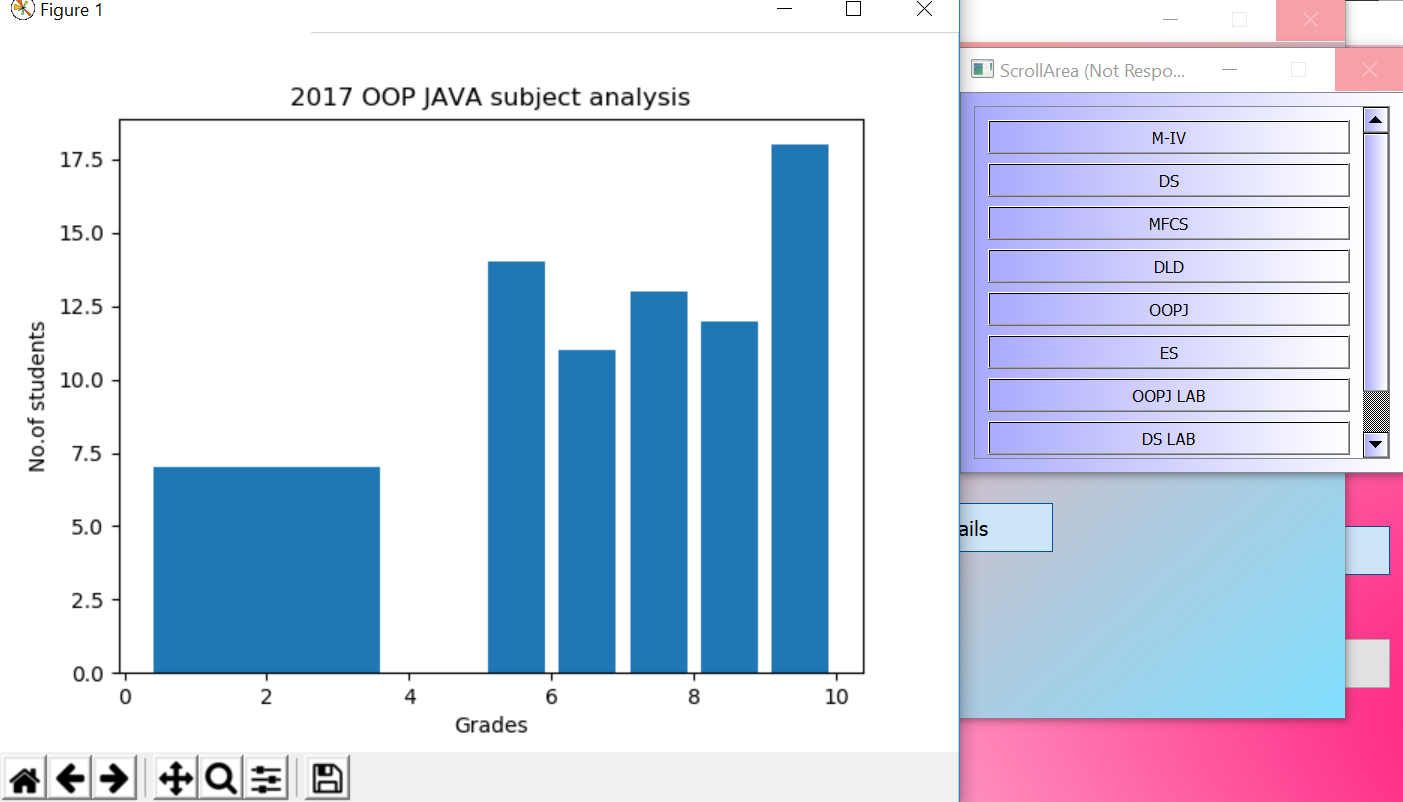


Fig 5.8: Subject Graphs

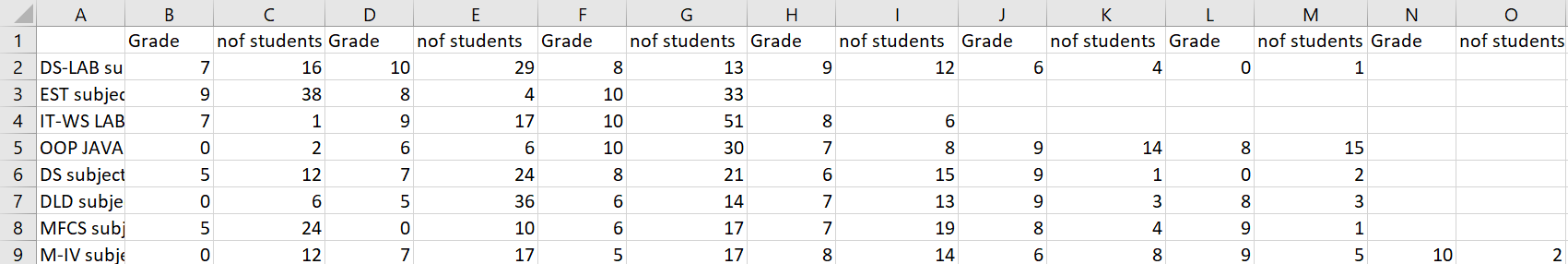


Fig 5.9: Section Grades

Fig 5.8 is displayed on choosing subject details button. It displays subject wise grades in a graphical format. The grade results are sent into the excel sheet as shown in the fig 5.9.

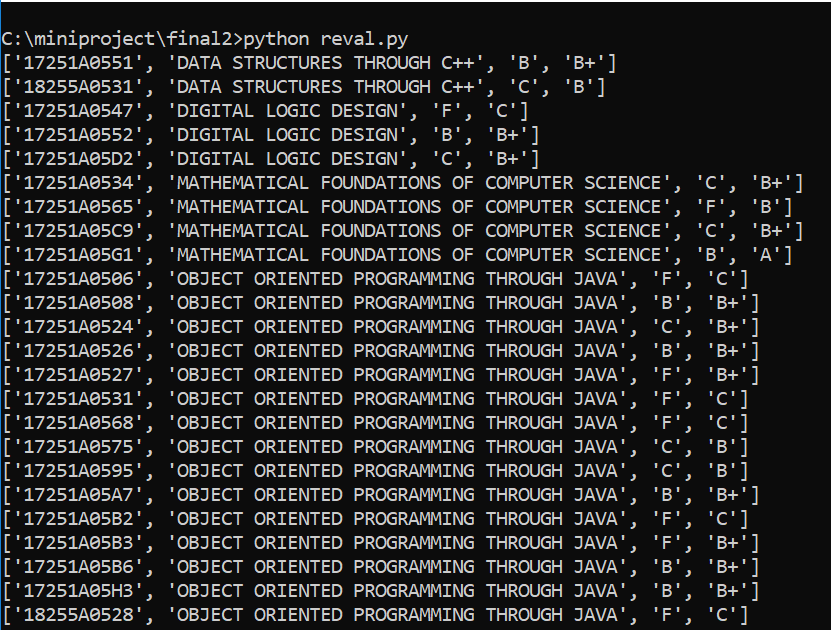


Fig 5.10 Re-evaluation Result

Fig 5.10 is displayed by the system when user chooses reval button in the main screen. It compares to excel sheets and displays the updated student details and results are sent to a new excel sheet.

# **6. CONCLUSIONS AND FUTURE ENHANCEMENTS**

## **6.1 Conclusion**

The system has achieved at least some of the automation and few difficulties are solved. The model is built in user-friendly manner. Analysis of results is done to display grade wise result of individual subject, Faculty results and re-evaluation results. The system can be easily used in college for student result analysis. It reduces time, which required for manual calculation. This system helps to calculate result fast so it optimizes the work force.

## **6.2 Future Scope**

Previously, data used to be inserted manually to analyze result. However, currently the project supports excel (.xlsx) files for extraction of data. The future scope is that data can be fetched, parsed in other formats like doc, csv, odt, etc. Visualization can be provided to represent data in graphical format. Various representations like pie chart, graph like histogram, etc. The latest technologies like python support various visualizations for analysis of result.

## **6.3 References**

1. <https://doc.lagout.org/programmation/python/Programming>
2. <https://www.tutorialspoint.com/sql/>
3. <https://www.python.org/>
4. <https://www.tutorialspoint.com/pyqt/pyqt_using_qt_designer.htm>
5. <https://www.sqlite.org/index.html>
6. <https://www.tutorialspoint.com/python_pandas/index.htm>
7. [https://www.irjet.net/archives/V5/i4/IRJETV5I4572.pdf 7](https://www.irjet.net/archives/V5/i4/IRJETV5I4572.pdf%207)
8. <https://numpy.org/>
9. <https://github.com/baoboa/pyqt5/blob/master/pyuic/uic/pyuic.py>
10. <https://riverbankcomputing.com/software/pyqt/intro>