A Major Project stage 2 Report

On

AUTOMATIC ASSESSMENT OF COMMUNICATION SKILLS IN NON-CONVENTIONAL INTERVIEW SETTINGS

Submitted to ***CMREC, HYDERABAD***
In Partial Fulfillment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted
By

Jangili Anusha	**(208R1A0524)**
G.Vineeth Kumar	**(208R1A0521)**
Anwar Hussain	**(208R1A0535)**
Damalla Jashwanth	**(208R1A0515)**

Under the Esteemed
guidance of

Mr. Y. Shyam Sundar

Assistant Professor, Department of CSE

Department of Computer Science & Engineering
CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE, NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)
2023-2024

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS

(Accredited by NBA,Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)
Kandlakoya, Medchal Road, Hyderabad-501401)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **"AUTOMATIC ASSESSMENT OF COMMUNICATION SKILLS IN NON-CONVENTIONAL INTERVIEW SETTINGS"** is a bonafide work carried out by

JANGILI ANUSHA	**(208R1A0524)**
G.VINEETH KUMAR	**(208R1A0521)**
ANWAR HUSSAIN	**(208R1A0535)**
DAMALLA JASHWATH	**(208R1A0515)**

in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this Major project Stage2 have been verified and are found to be satisfactory. The results embodied in this Major project Stage2 have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Major Project Coordinator**	**Head of the Department**
Mr. Y. Shyam Sundar	**Mrs. G. Sumalatha**	**Dr. Sheo Kumar**
Assistant Professor	Associate Professor	Professor & H.O.D
CSE Department	CSE Department	CSE Department
CMREC	CMREC	CMREC

DECLARATION

This is to certify that the work reported in the present Major project stage2 entitled **"AUTOMATIC ASSESSMENT OF COMMUNICATION SKILLS IN NON-CONVENTIONAL INTERVIEW SETTINGS"** is a record of bonafide work done by us in the Department of Computer Science and Engineering, CMR Engineering College, JNTU Hyderabad. The reportsare based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this Major project stage2 report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Jangili Anusha	**(208R1A0524)**
G.Vineeth Kumar	**(208R1A0521)**
Anwar Hussain	**(208R1A0535)**
Damalla Jashwanth	**(208R1A0515)**

ACKNOWLEDGMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr.Sheo Kumar**, HOD, **Department of CSE, CMR Engineering College** for their constant support**.**

We are extremely thankful to **Mr. Y. Shyam Sundar,** Assistant Professor, Internal Guide, Department of CSE, for his constant guidance, encouragement and moral support throughout the project.

We will be failing in duty if We do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

We thank Mrs.G.Sumalatha, **Associate Professor, CSE Department,** Major Project Coordinator for his constant support in carrying out the project activities and reviews.

We express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, We are very much thankful to my parents who guided us for every step.

Jangili Anusha	**(208R1A0524)**
G.Vineeth Kumar	**(208R1A0521)**
Anwar Hussain	**(208R1A0535)**
Damalla Jashwanth	**(208R1A0515)**

CONTENTS

TOPIC **PAGENO**

ABSTRACT

Automatic assessment of communication skills in non-conventional interview settings refers to the use of technology to evaluate and analyze the communication abilities of individuals during job interviews or similar interactions, especially in unconventional or remote settings. With the rise of digital communication platforms and remote work opportunities, assessing communication skills in non-traditional interview settings has become increasingly important for employers to identify suitable candidates. Traditional systems for assessing communication skills in job interviews relied heavily on human judgment. Interviewers assessed candidates based on their verbal communication, body language, and responses to questions. However, this approach is subjective and can be influenced by interviewer bias. In addition, traditional face-to-face interviews have limitations, especially when it comes to assessing candidates' communication skills in diverse contexts. Non-conventional interview settings, such as virtual interviews, phone calls, or chat- based interactions, require different communication skills compared to in-person meetings. With the advent of technology, there has been a shift towards using automated systems that employ natural language processing, sentiment analysis, and machine learning algorithms to provide objective and data-driven assessments. Therefore, there is a need for automated tools and systems that can accurately evaluate candidates' communication abilities in these non-traditional settings. This research proposes a comparative study on various technologies designed to predict communication skills in interview candidates. These predictions are based on video, audio, and written essay questions, and answers. Machine learning algorithms, specifically XGBOOST and deep learning methods, are employed in all prediction models.

LIST OF FIGURES

1.INTRODUCTION

1.1 Introduction:

Communication is the aptitude to proficiently and expediently transmit information to others. Proficiency in communication skills can have a positive impact on various domains of life, encompassing both professional environments and social contexts, among others. Proficiency in several forms of communication is an essential prerequisite for any employment position held by a knowledge worker. There exist various forms of communication, including spoken, written, and non-verbal. Communication skills encompass more than just face-to-face interactions and verbal communication. The proficiency in producing coherent and impactful written communication is equally essential. The automation of the recruiting process has been made possible by the recent emergence of machine learning in the sector, sometimes referred to as recruitment analytics or hiring analytics. The utilization of social recruitment, virtual evaluations, and video interviews is regarded as the prospective direction for the hiring procedure. Several automated talent assessment platforms, such as Teleview, Hirevue, and Sonru, are currently receiving significant attention. These platforms provide widespread access to interviews and screening processes. However, there are limited number of urgent inquiries that require attention. How do candidates fare in automated environments? What are the observed variations in the behavior and characteristics of the applicants across different settings? The present study examines responses to the a fore mentioned inquiries. In order to accomplish this objective, we have gathered interviews from individuals in three distinct contexts: video interviews, written interviews, and short essays conducted through a specially designed online interface. Subsequently, a range of analyses are conducted on the data, followed by the development of a predictive model aimed at automating the evaluation of participants communication abilities.

Objective:

The objective of this study is to evaluate the viability of automated assessment solutions.

The aim of this study is to examine the behavioral and performance.

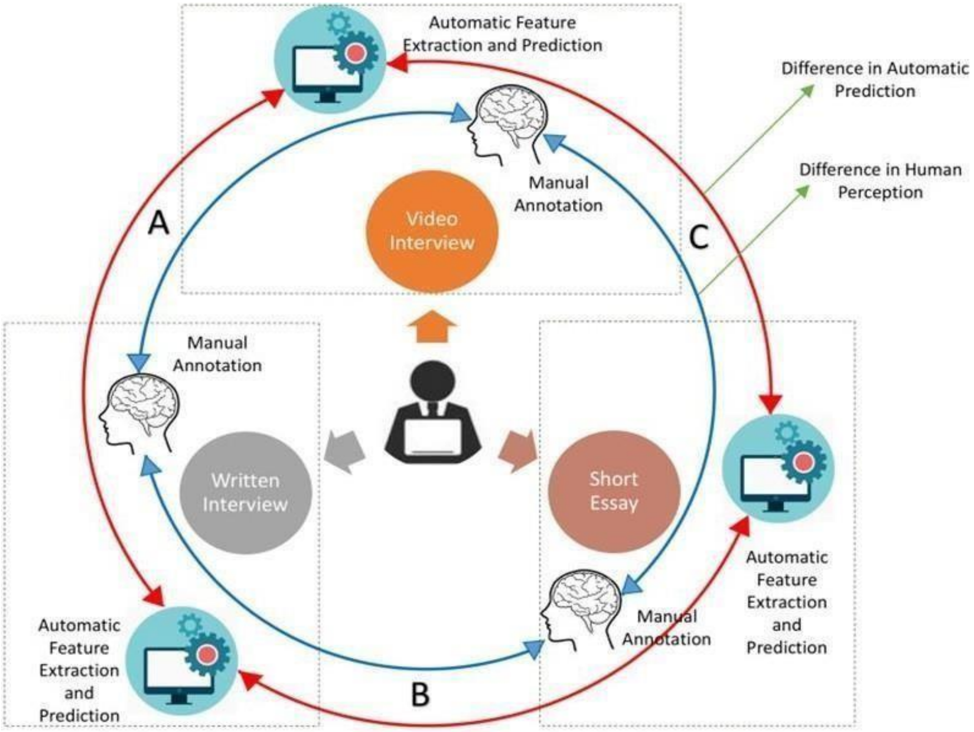

Automatic Feature Extraction and Prediction
Difference in Automatic Prediction
Difference in Human Perception
Manual Annotation
Video Interview
A
C
Manual Annotation
Written Interview
Short Essay
Automatic Feature Extraction and Prediction
Manual Annotation
Automatic Feature Extraction and Prediction
B

Figure1: Non-conventional Interview System. Bluelines-Differences in human perception, redlines - Differences in automatic features and prediction, (A) Between the Video and Written interview (B) Between Written interview and Short Essay (C) Between Video interview and Short Essay

1.2 Purpose of the Project:

Traditional face-to-face interviews have limitations, and with the rise of remote work and digital communication platforms, there is a growing need for effective methods to assess communication skills in non-conventional interview settings. Evaluating candidates objectively and accurately is crucial for employers to identify suitable candidates, especially in diverse and remote work environments.

Problem Definition

The problem lies in the subjective nature of traditional communication skills assessment methods. Human judgment, which is susceptible to biases, often leads to inconsistent evaluations. Additionally, non-conventional interview settings, such as virtual interviews phone calls, require different communication skills compared to in-person meetings, making it challenging to assess candidates fairly.

Significance

I. Objective Evaluation: Automated assessment tools offer objective evaluations, eliminating human bias and ensuring fairness in the hiring process.

II. Efficiency: Automation speeds up the evaluation process, allowing employers to assess a large number of candidates efficiently, which is especially valuable in high-volume recruitment scenarios.

III. Cost-Effectiveness: Automating the assessment process reduces the resources spent on manual evaluations, making it a cost-effective solution for businesses.

IV. Enhanced Decision Making: Accurate assessment of communication skills leads to better hiring decisions, ensuring that candidates with the right skills and abilities are selected for the job.

V. Adaptability: The set tools can be tailored to assess communication skills relevant to specific job roles or industries, ensuring a customized evaluation approach.

Applications

I. Recruitment and Hiring: Employers can use automated assessment tools during the recruitment process to evaluate candidates' communication skills, ensuring that they align with the job requirements.

II. Training and Development: Identifying communication skill gaps in existing employee scan aid in designing targeted training programs, enhancing the overall communication abilities of the workforce.

III. Education: Educational institutions can utilize similar technologies for assessing students' communication skills, providing valuable feedback for improvement.

IV. Research and Analysis: Automated assessment tools can be used in research studies to analyze communication patterns, aiding in sociolinguistic research and related fields.

V. Customer Service: Businesses employing customer service representatives can assess their communication skills, ensuring excellent interactions with clients and customers

1.3 Existing System

In the traditional system, assessing communication skills during job interviews or similar interactions relied heavily on human judgment. Interviewers evaluated candidates based on verbal communication, body language, responses to questions, and overall presentation during face-to-face interviews. These assessments were typically subjective and based on the interviewer's personal biases and perceptions. The limitations of the traditional system, including subjectivity, lack of scalability, inconsistency, and the inability to assess diverse communication skills, highlight the need for more objective, scalable, and data-driven approaches to communication skills assessment in modern recruitment processes.

Limitations

1. Subjectivity and Bias:

   - Traditional assessments are highly subjective and can be influenced by the interviewer's biases, leading to inconsistent and unfair evaluations.
   - Interviewers might unintentionally favor candidates who share similar backgrounds, attitudes, or communication styles, leading to biased judgments.

2. Limited Context:
   - Face-to-face interviews often provide a limited context for evaluating communication skills.
   - Candidates may perform differently in real workplace scenarios or remote communication settings, which are becoming increasingly common in the digital age.

3. Scalability Issues:

   - Manual assessments are time-consuming and resource-intensive, especially in large-scale recruitment processes with numerous applicants.
   - Scaling up traditional assessments to accommodate a high volume of the candidates can be lead to delays in the hiring process.

4. Lack of Consistency:

- Different interviewers may have varying evaluation criteria and standards.
- Lack of consistency makes it challenging to compare candidates objectively and hampers fair decision-making.

5. Limited Feedback:

- Traditional assessments often provide limited feedback to candidates, offering vague insights into their performance.
- Candidates may not receive specific details about their strengths and weaknesses, hindering their ability to improve their communication skills.

6. Inability to Assess Diverse Skills:

- Traditional methods may not effectively assess diverse communication skills required for various job roles, such as remote collaboration, written communication, or virtual presentation abilities.

7. Human Error and Fatigue:

- Interviewers, like any human, are prone to error and can become fatigueddduring long interview sessions, affecting the accuracy of their assessments.

8. Limited Data for Analysis:

- Traditional methods do not generate structured data that can be analyzed for patterns or trends.
- Lack of data-driven insights hampers the ability to make informed decisions or identify areas for improvement in the assessment process.

1.4 Proposed System:

The proposed automated system addresses the limitations of traditional system by leveraging technology and machine learning algorithms to provide fair, efficient, and insightful evaluations of candidate’s communication abilities in non-conventional interview settings.

Overview

The proposed system enables the automatic assessment of communication skills in non-conventional interview settings, allowing users to receive feedback on their visual expressions, spoken content, and written communication skills. The project integrates computer vision, audio processing, and natural language processing techniques to perform these assessments and provide valuable insights to the users.

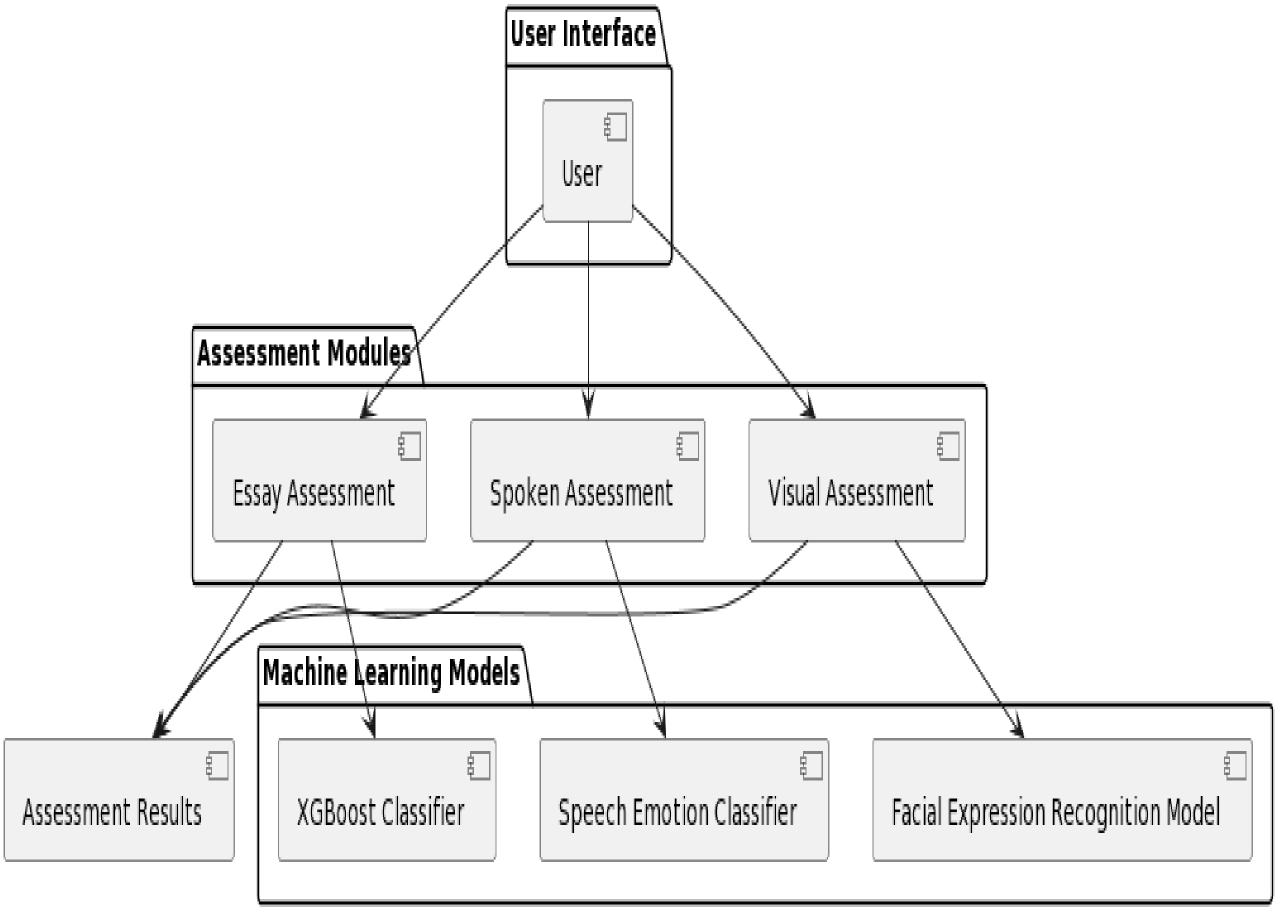


Figure 1.4.1 Proposed system architecture.

1. User Interface:

   - The project starts with a graphical user interface (GUI) created using the Tkinter library in Python.
   - The interface provides options for the user to perform different types of assessments: visual, spoken, and essay assessments.

2. Loading Assessment Models:

   - When the user clicks the "Generate & Load Assessment Model" button, the system loads pre-trained machine learning models necessary for assessments.
   - These models include a facial expression recognition model, a speech emotion classifier, and an XGBoost classifier for text data.

3. Visual Assessment:

   - ✓ If the user chooses the "Visual Interview Assessment" option:
     - The system activates the computer's camera to capture video frames.
     - It uses a pre-trained facial expression recognition model to detect facial expressions in the video frames.
     - The system calculates the confidence and confusion percentages based on the detected expressions.
     - The results are displayed back to the user, indicating their confidence and confusion-levels during the visual assessment.

4. Spoken Assessment:

   - ✓ If the user selects the "Spoken Interview Assessment" option:
     - The user is prompted to select an audio file containing spoken content.
     - The system extracts relevant features from the audio file.
     - It utilizes a pre-trained speech emotion classifier to predict the spoken emotion, categorizing it as confident or confused.
     - The system informs the user about the prediction result, indicating whether the spoken content sounds confident or confused.

5. Essay Assessment:

- ✓ If the user opts for the "Written & Short Essay Assessment":
    - o The user enters written content in the provided text box.
    - o The system processes the text, cleaning it by removing punctuation, stop words, and stemming/lemmatizing words.
    - o The cleaned text is vectorized using a TF-IDF vectorizer.
    - o The pre-trained XGBoost classifier predicts the communication skill level based on the processed essay content.
    - o The system displays the prediction score to the user.

6. Displaying Results:

- o The assessment results, including confidence and confusion percentages from the visual assessment, spoken assessment prediction, and essay assessment prediction score, are displayed in the GUI.
- o For visual assessment, the system displays the count of confident and confused facial expressions.
- o For spoken assessment, the system informs the user whether their speech sounded confident or confused.
- o For essay assessment, the system shows the predicted communication skill level based on the written content.

7. Data Flow and Processing:

- o The project utilizes machine learning models and techniques to process different types of data:
- o Visual assessment involves processing video frames using a facial expression recognition model.
- o Spoken assessment extracts features from audio files and uses a speech emotion classifier.
- o Essay assessment processes text data, cleaning and vectorizing it before prediction.

8. User Interaction and Feedback:

- The user interacts with the system through the GUI, inputting data or selecting files for assessment.
- The system provides feedback to the user, informing them of the assessment results and predictions.

2.LITERATURE SURVEY

The development of an automated interview scoring system for evaluating communication skills involves the integration of multiple disciplines, encompassing research in areas such as computer-assisted education, automatic video assessment, automated content scoring (including short answer grading and essay scoring), and affective computing. The subsequent paragraphs provide a concise overview of the research conducted in these specific domains. Recently, there has been a significant surge in research interest surrounding automated video assessment. Numerous social variables have been examined.

Bartica et al. proposed an automated method for predicting personality traits by extracting auditory features and manually extracting visual features. Biel et al. conducted a study in which they made predictions about the Big Five personality traits based on facial expressions of emotions observed in internet chat videos. Nguyen et al. provided a computational framework aimed at predicting liability in real employment interviews through the extraction of non-verbal cues exhibited by both the interviewee and the interviewer. In their study, Rasipuram et al. make a prediction on the effectiveness of audio-visual cues in assessing communication skills during employment interviews. In a separate investigation, the MIT Inter-view dataset was employed to forecast the comprehensive work performance, encompassing overall rating and recommended hiring, as well as 14 distinct social attributes such as excitement, friendliness, and engagement. The study yielded encouraging outcomes. Rasipuram and colleagues propose the utilization of a dual dataset comprising of asynchronous and face-to-face interviews in order to analyze and compare the disparities between them. Their objective is to estimate the level of communication competence based on these interviews. There are a plethora of resources available to facilitate the development and enhancement of social and presenting aptitudes. Several examples of automated systems designed to improve social skills include MACH, Rhema, ROC Speak, Auto manner, and the Automated Social Skills Trainer.

The variable under consideration is communication skill, which is deemed crucial for achieving success in any career interview. Our research article not only centers on the development of oral communication skills, but also involves a comparative analysis using a non-conventional interview . Our objective is to determine the viability of utilizing this approach as a potential substitute for interface-based video interviews, particularly in situations where infrastructure limitations, such as bandwidth constraints, are prevalent. The domain of automatic assessment of textual communication in interview settings has received limited attention in academic research. It can be matched with the domain of automated content assessment. The algorithmic assessment of natural language answers has been a subject of research since the early contributions of Page. Burrows et al. provide an extensive examination of the research conducted in the field of automated grading of brief answers. C-rater is an early system developed by ETS that is designed to assess contrived responses in relation to certain prompts. According to the cited source, the evaluation of constructed responses is mostly based on their substance rather than the level of writing proficiency. Several contemporary systems, including a few recent ones, rely on manually crafted patterns to evaluate the accuracy of responses. Ramachandran et al. provide a methodology for the automated extraction of patterns from the reference answers. Additional approaches include the utilization of text-to-text similarity metrics to compare the responses provided by students with the standard or reference replies.

Higgins et al. extend their investigation beyond the utilization of manual features that are relevant to the query and delve into the examination of features that are informed by syntax. In the given task, the responses to behavioral inquiries are unknown and unique for every participant, making it impractical to train on predefined patterns. Furthermore, given the inherent subjectivity within this field, it is not feasible to establish definitive or universally accepted answers. The evaluation of texts by automatic essay scoring is a significant domain of study that has been extensively investigated. Project Essay Grader emerged as a pioneering system in the field of automated text evaluation, focusing exclusively on assessing the writing quality. Several further models, such as the E-rater, Intelligent Essay Assessor, Bayesian Essay Test Scoring System, and IntelliMetric, have been developed and proven to be successful. The E-rater system, developed by ETS, employs regression models and natural language processing techniques to extract features and attain an efficiency level ranging from 87% to 94%. These scores were compared to human-based evaluations, which were considered as the benchmark for accuracy.

3.SOFTWARE REQUIREMENT ANALYSIS

3.1 Problem Specification:

Problem Statement:

The problem of active online learning for social media analysis in crisis management involves developing a system that can autonomously select and label the most informative posts or messages related to a crisis. The goal is to filter out noise and focus on the data that is most likely to provide critical insights for crisis response and management. By leveraging machine learning algorithms, this approach can dynamically select and label data points.

SDLC:

SDLC stands for Software Development Life Cycle. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software. SDLC is the process consisting of a series of planned activities to develop or alter the software products.

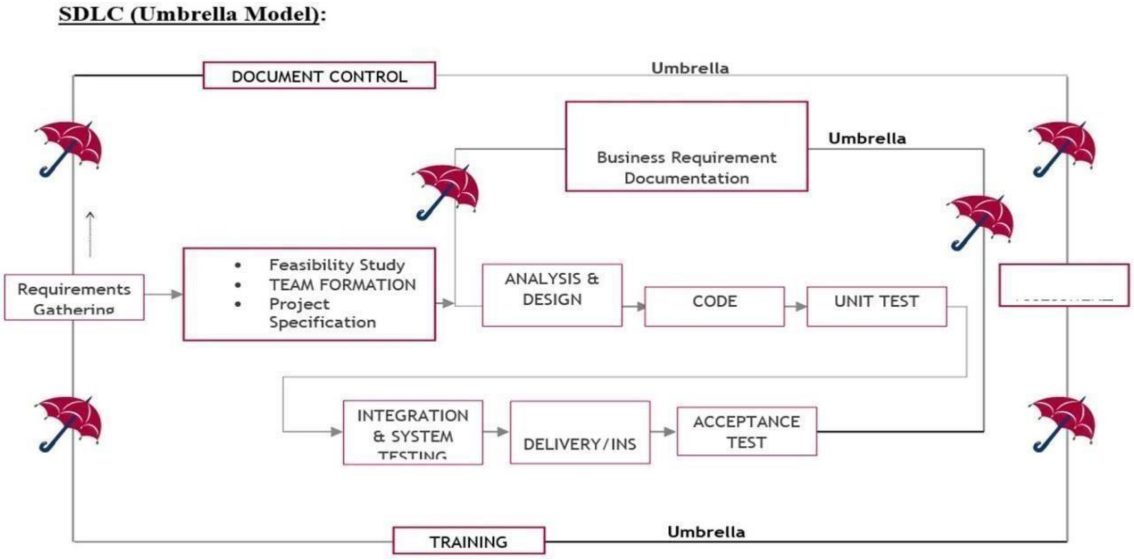


Fig: 3.1.1 SDLC block diagram

SDLC block diagram

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

Stages in SDLC:

- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

Requirement Analysis and Design:

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

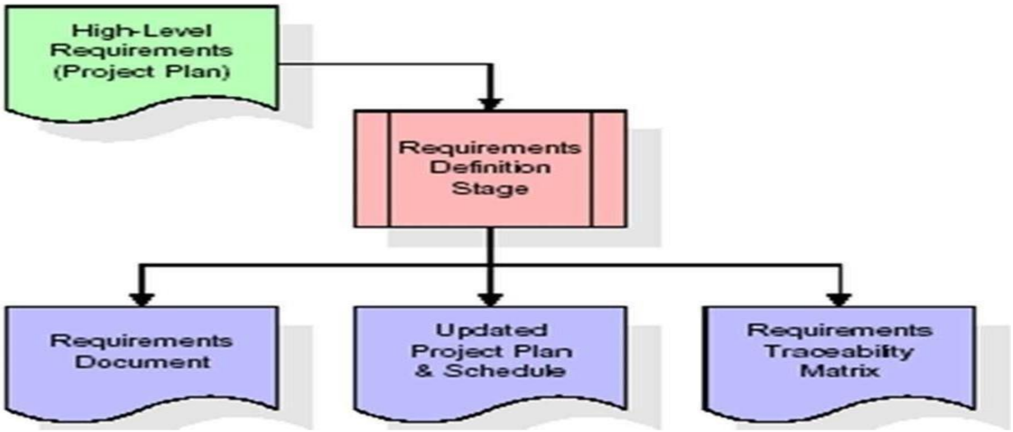

High-Level Requirements (Project Plan)
Requirements Definition Stage
Requirements Document
Updated Project Plan & Schedule
Requirements Traceability Matrix

Fig: 3.1.2 Requirement Analysis and Design

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirement Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development life cycle are formally connected to the components developed in prior stages. In the requirements stage, theRTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

Theoutputsoftherequirementsdefinitionstageincludetherequirements document, the RTM, and an updated project plan.

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

Analysis Stage:

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

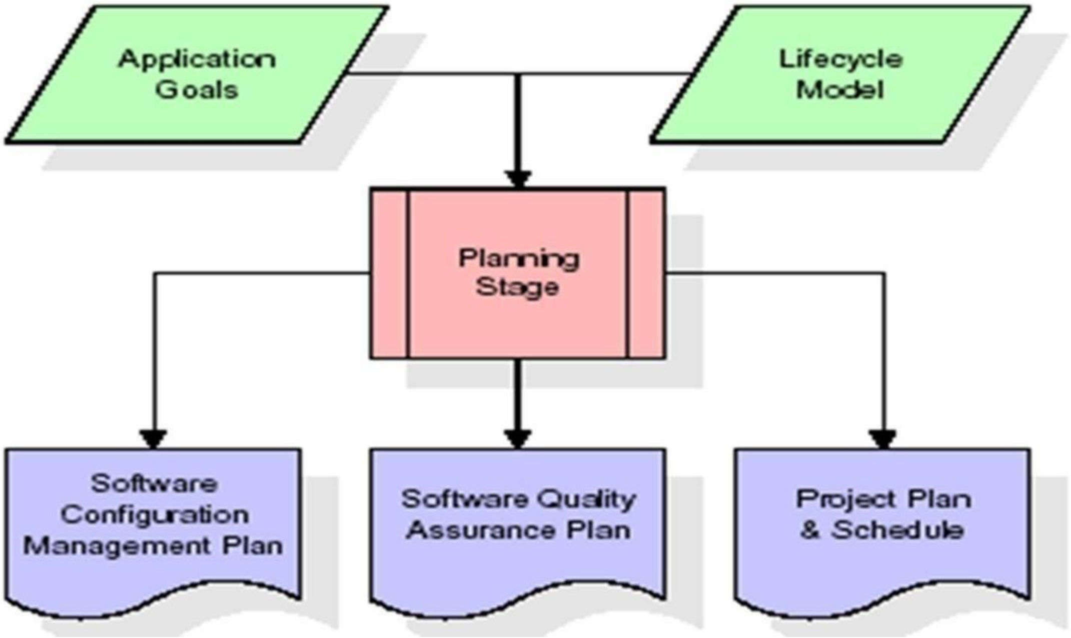

Application Goals
Lifecycle Model
Planning Stage
Software Configuration Management Plan
Software Quality Assurance Plan
Project Plan & Schedule

Fig: 3.1.3 Analysis stage

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

Designing Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudocode and a complete entity relationship diagram with a full data dictionary.

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

Development (Coding) Stage:

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

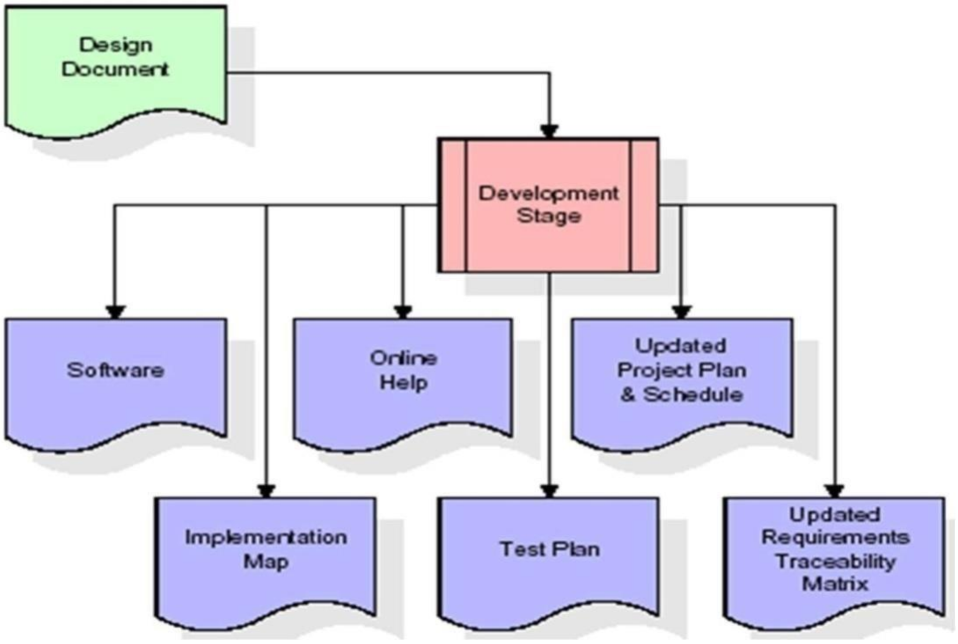


Fig: 3.1.4 Development(coding) stage

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

Integration & Test Stage:

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

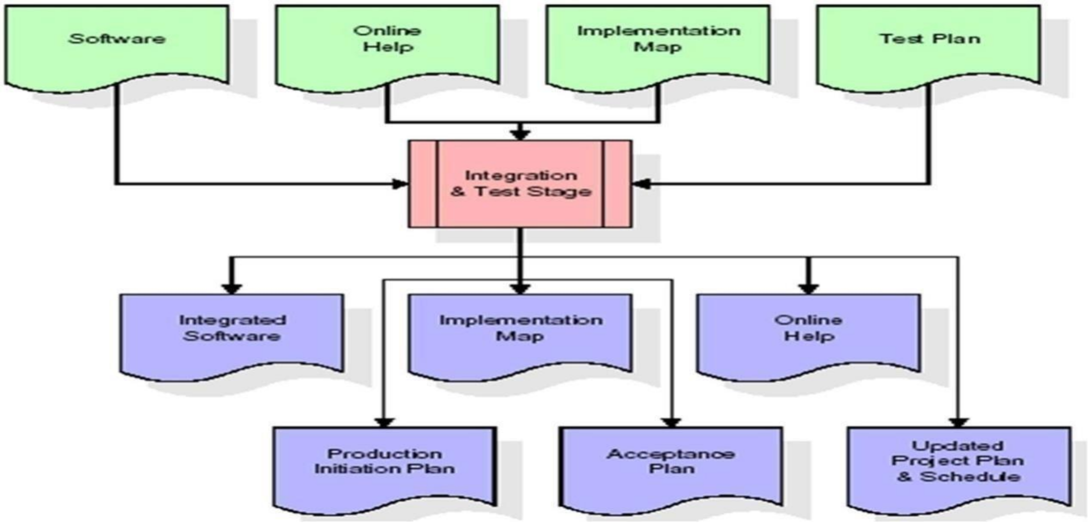


Fig:3.1.5 Integration and Test stage

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

Installation & Acceptance Test:

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded on to the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer. After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

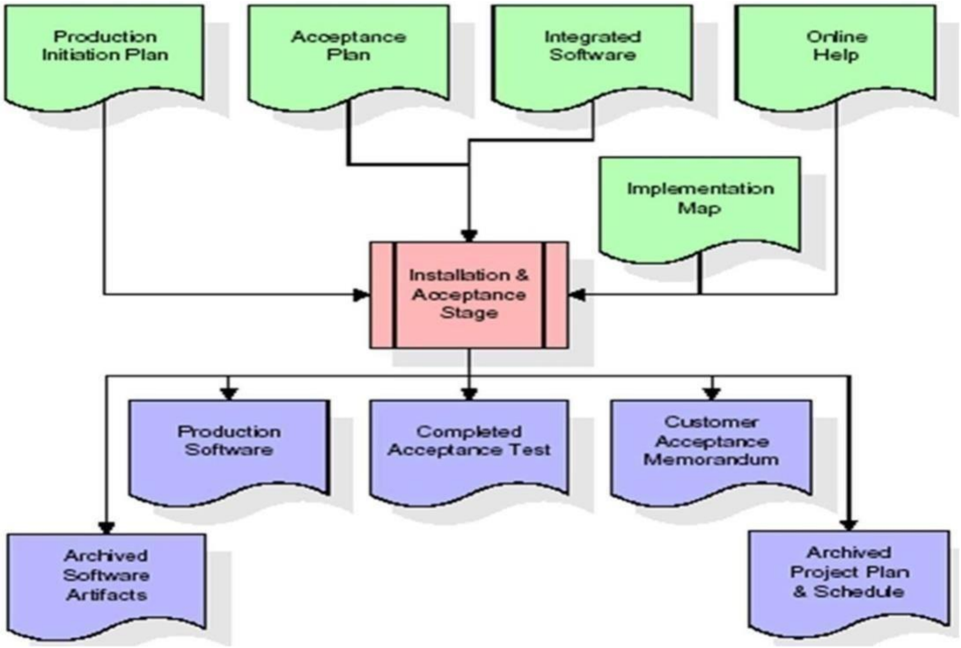

Production Initiation Plan
Acceptance Plan
Integrated Software
Online Help
Implementation Map
Installation & Acceptance Stage
Production Software
Completed Acceptance Test
Customer Acceptance Memorandum
Archived Software Artifacts
Archived Project Plan & Schedule

Fig: 3.1.6 Installation and Acceptance test

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

Maintenance:

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

3.2 Functional Requirements

Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within the organization.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

Output Definition

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

Input Design

Input design is a part of overall system design. The main objective during the input design is:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user

Input Stages

The main input stages can be listed as below:

- Datarecording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission

- Data validation
- Data correction

Input Types

Itis necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue.

Input Media

At this stage choice has to be made about the input media.
To conclude about the input mediaconsideration has to begiven to

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

Error Avoidance

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

Error Detection

Even though every effort is made to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

Data Validation

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

User Interface Design

It is essential to consult the system users and discuss the needs while designing the user interface:

User Interface Systems Can Be Broadly Classified As:

- User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
- Computer initiated interfaces in the computer-initiated interfaces the computer guides the progress of the user/computer dialogue.
- Information is displayed and the user response of thecomputer takes action or displays further information.

User Initiated Interfaces

User initiated interfaces all into two approximate classes:

- Command driven interfaces: In this type of interface, the user inputs commands or queries which are interpreted by the computer.
- Forms oriented interface : It is chosen because it is best choice.

Computer-Initiated Interfaces

The following computer –initiated interfaces were used:

- The menu system for the user is presented with a list of alternatives.
- The user chooses one of the alternatives.
- Questions–answer type dialog system where the computer asks question and takes action.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

Error Message Design

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed. This application must be able to produce output at different modules for different inputs.

Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

3.3 Non-Functional Requirements

The non-functional requirements specify the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non- functional standards that are critical to the success of the software system. Example of non- functional requirement, “how fast does the website load?” Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

- Usability requirement
- Service ability requirement
- Manage ability requirement
- Recoverability requirement
- Security requirement
- Data Integrity requirement
- Capacity requirement
- Availability requirement
- Scalability requirement
- Interoperability requirement
- Reliability requirement
- Maintainability requirement
- Regulatory requirement
- Environmental requirement

Advantages of Non-Functional Requirement

Benefits of Non-functional testing:

- The non-functional requirements ensure the software system follow legal and compilation rules.
- They ensure the reliability, availability, and performance of the software system.
- They ensure good user experience and as of operating the software.

Disadvantages of Non-functional requirement

Drawbacks of non-function requirement are:

- Non-functional requirement may affect the various high-level software sub system.
- They require special consideration during the software architecture/high-level design phase which increases costs.
- Their implementation does not usually map to the specific software sub-system,
- It is tough to modify non-functional once you pass the architecture phase.

3.4 Feasibility Study

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

Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well with in the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement.

Social Feasibility

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

4.SYSTEMREQUIREMENTSSPECIFICATION

4.1 Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

- PythonIDLE3.7version (or)
- Anaconda3.7(or)
- Jupiter(or)
- Google Collab

4.2 Hardware Requirements

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

• Operating system	:	Windows, Linux
• Processor	:	minimuminteli3
• Ram	:	minimum4 GB
• Hard disk	:	minimum250GB

4.3 SELECTEDSOFTWARE:

What is Python?

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms.
- Pythonprograms generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like–Google, Amazon, Face book, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries used for-

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by You Tube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frame works
- Multimedia

Advantages of Python

Let’s see how Python dominates over other languages.

1. Extensive Libraries

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

2. Extensible

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

3. Embeddable

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

4. Improved Productivity

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

5. IOT Opportunities

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

6. Simple and Easy

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

7. Readable

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

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Last, we will say that python is an interpreted language. Since statements are executed one by one.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support. The 2019Githubannualsurveyshowed us that Python has overtaken java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Macor Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform a data analysis and a machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the down sides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbon Nelle.

The reason it is not so famous despite the existence of Bryton is that it isn’t that secure.

3. Design Restrictions

As you know, Python is dynamically typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Under developed Database Access Layers

Compared to more widely used technologies like JDBC (Java Data Base Connectivity) and ODBC (Open Data Base Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary. This was all about the Advantages and Disadvantages of Python Programming Language.

Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great white space. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects.

How to Install Python on Windows and Mac

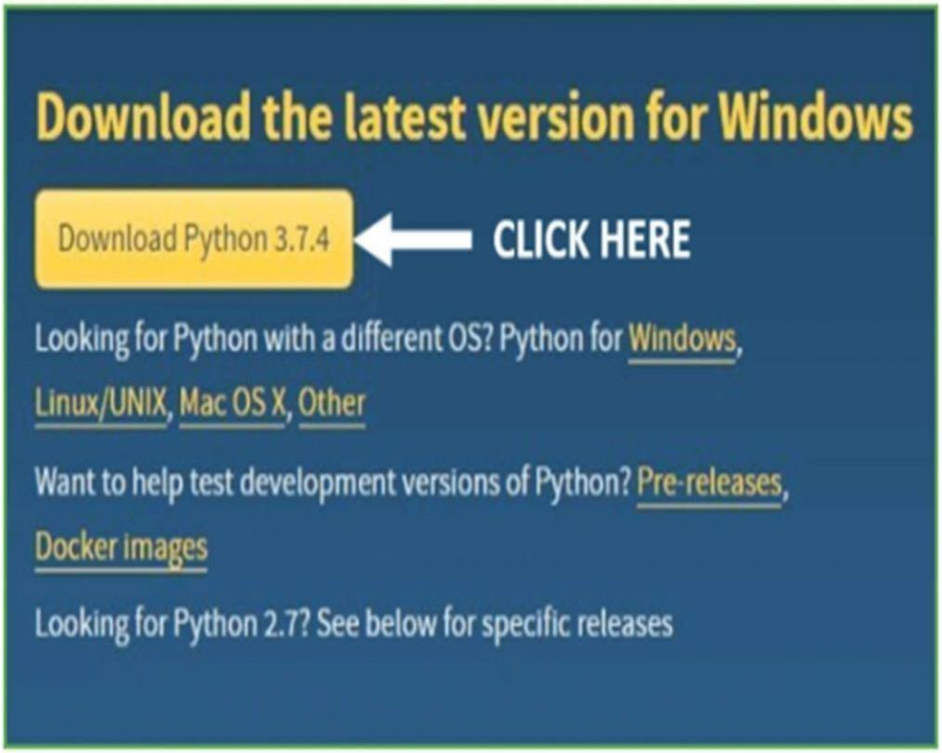
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other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

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

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows64-bit operating system. So, the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download The Python Creates there. The step so how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org



Download the latest version for Windows

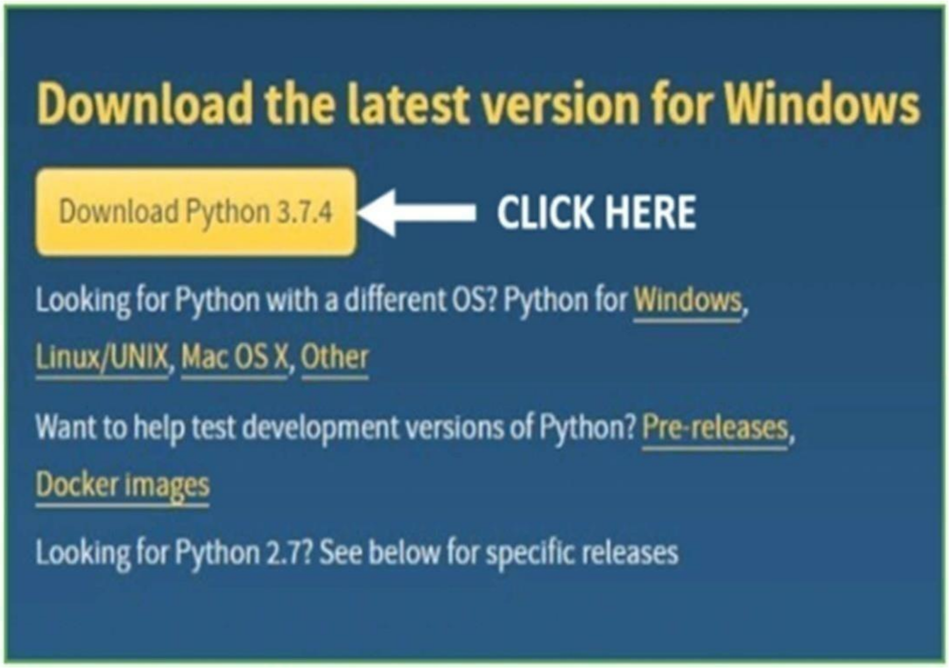
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

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

Step2: Click on the Download Tab.



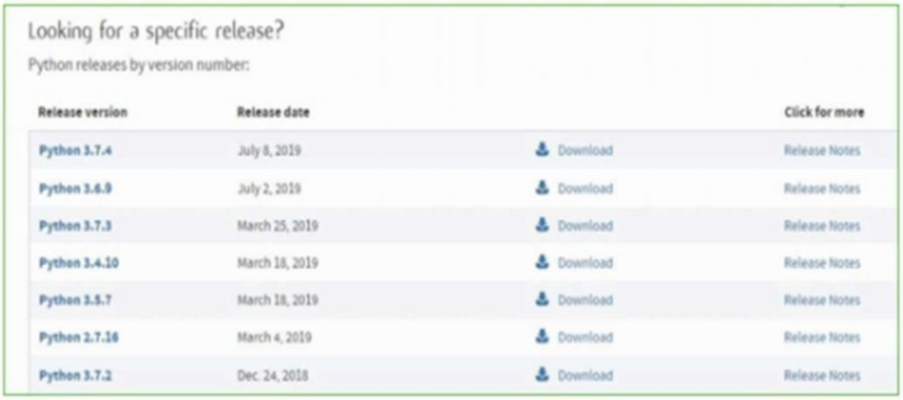
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

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



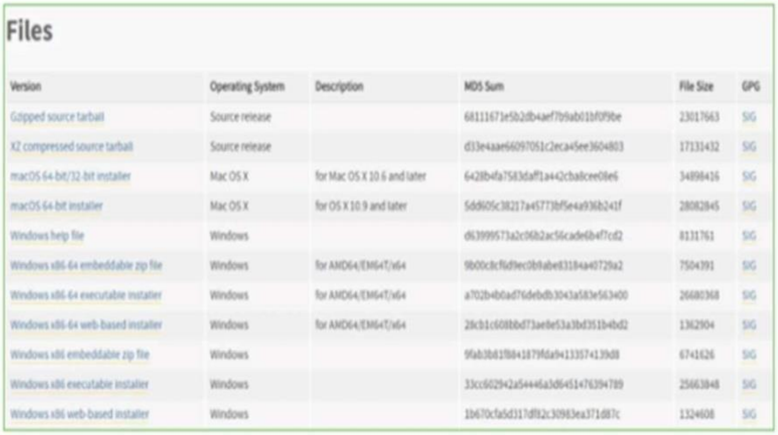
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step4: Scroll down the page until you find the Files option.

Step5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

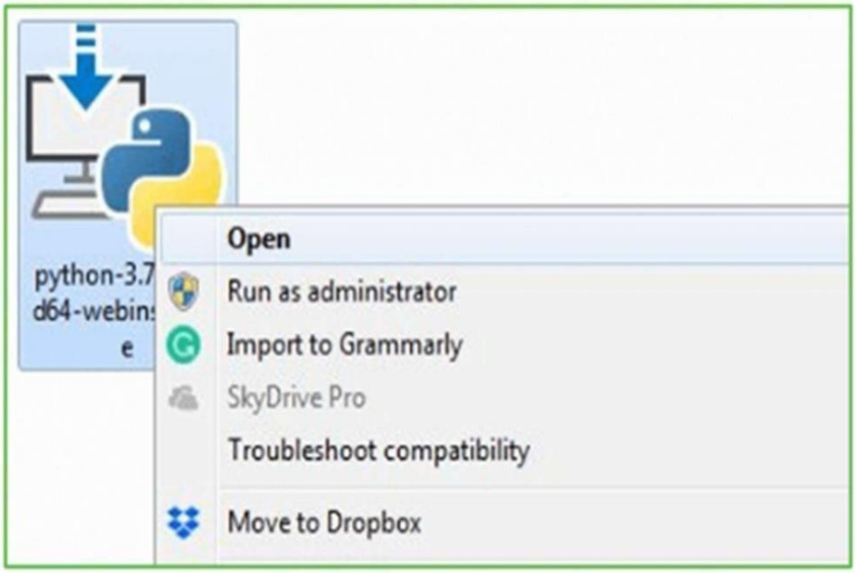
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be down loaded is completed. Now we move ahead with these second part in installing python i.e. Installation

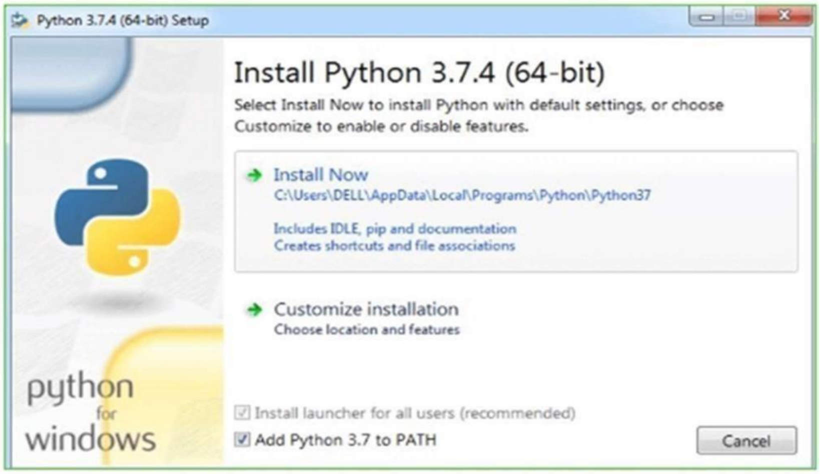
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

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



Step2: Before you click on Install Now, make sure to put a tick on Add Python3.7 to PATH.


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

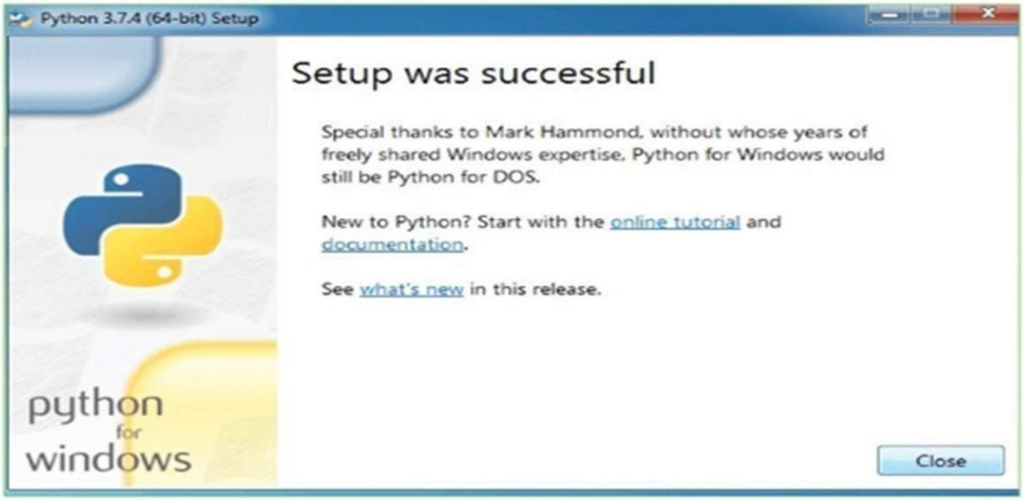
→ Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Step3: Click on Install NOW. After the installation is successful, Click on Close.



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

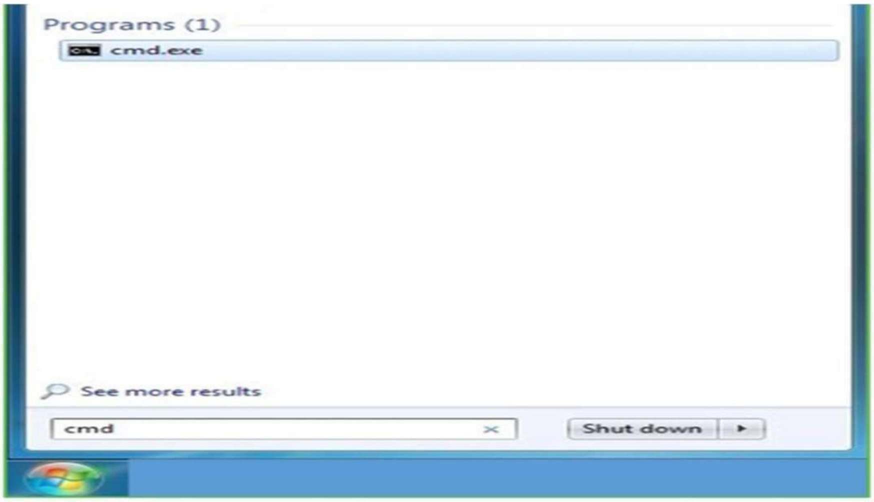
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

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

Verify the Python Installation

Step1: Click on Start

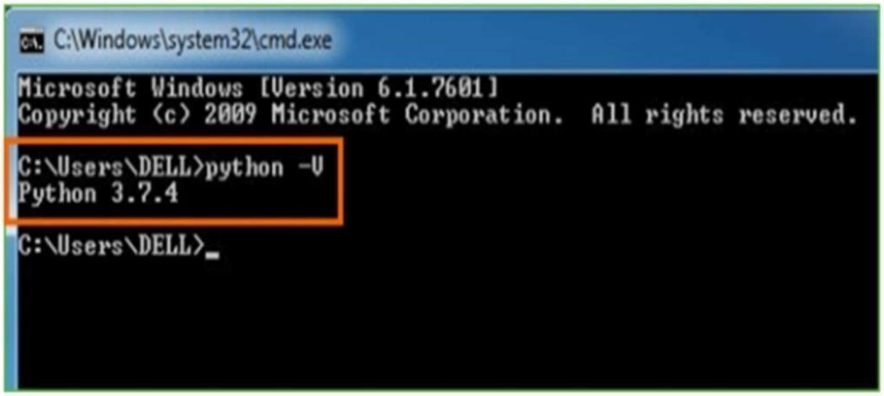
Step2: In the Windows Run Command, type “cmd”.



cmd

Step3: Open the Command promptoption.

Step4: Let us test whether the python is correctly installed. Type python–V and press Enter.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

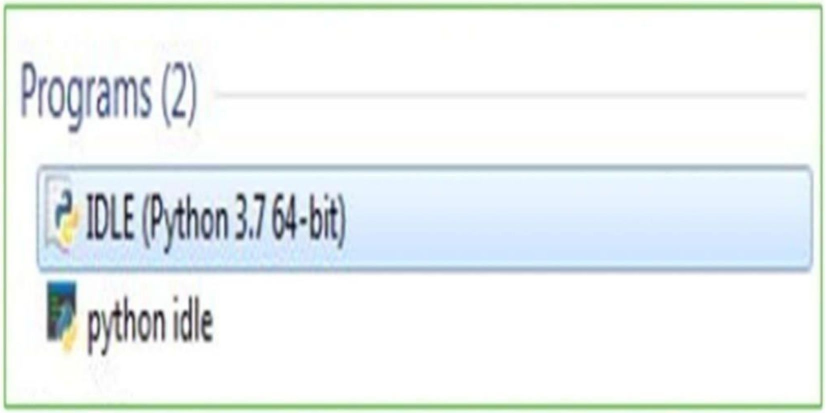
Step5:Youwillgettheansweras3.7.4

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

Check how the Python IDLE works

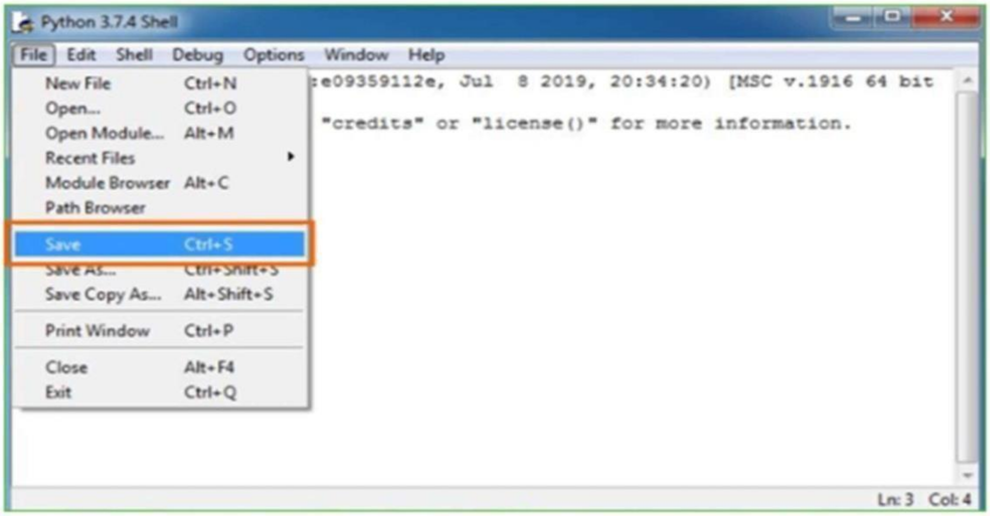
Step1: Click on Start

Step2: In the Windows Run command, type “pythonidle”.



Step3: Click on IDLE(Python3.764-bit) and launch the program

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

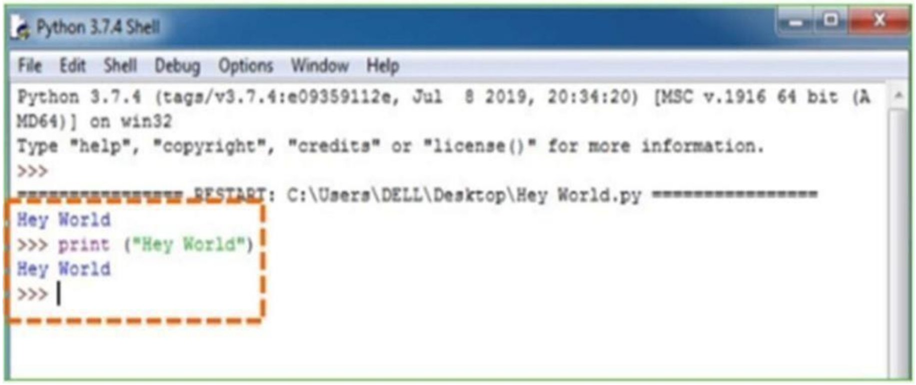


```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit

 "credits" or "license()" for more information.
```

Step5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step6: Now for e.g. enter print(“Hey World”)and Press Enter.



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (A
MD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:\Users\DELL\Desktop\Hey World.py ================
Hey World
>>> print ("Hey World")
Hey World
>>> |
```

You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

5.SOFTWARE DESIGN

5.1Data flow diagram

A Data Flow Diagram (DFD) is a visual representation of the flow of data within a system or process.It is a structured technique that focuses on how data moves through different processes and data stores within an organization or a system. DFDs are commonly used in system analysis and design to understand, document, and communicate data flow and processing.

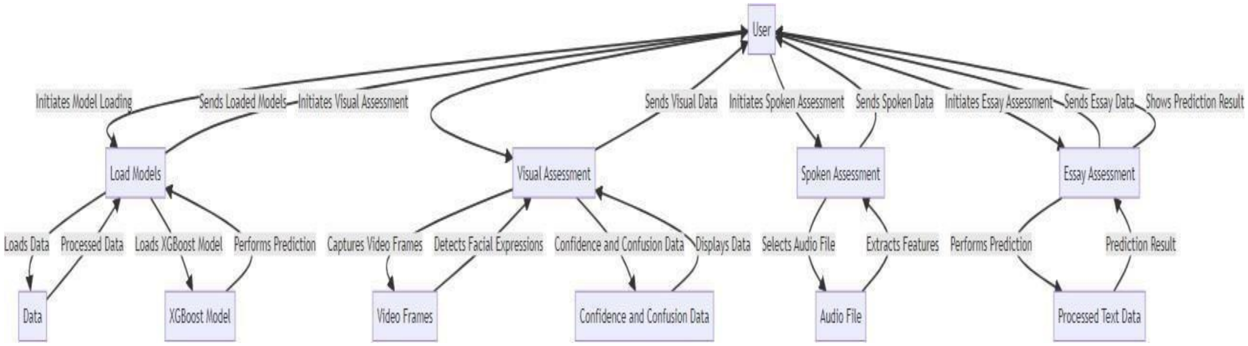


Fig: 5.1.1 Data flow diagram

5.2 UML diagram

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

The Unified Modeling Language Is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

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

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

Class Diagram

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an “is-a” or “has-a” relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed “methods” of the class. Apart from this, each class may have certain “attributes” that uniquely identify the class.

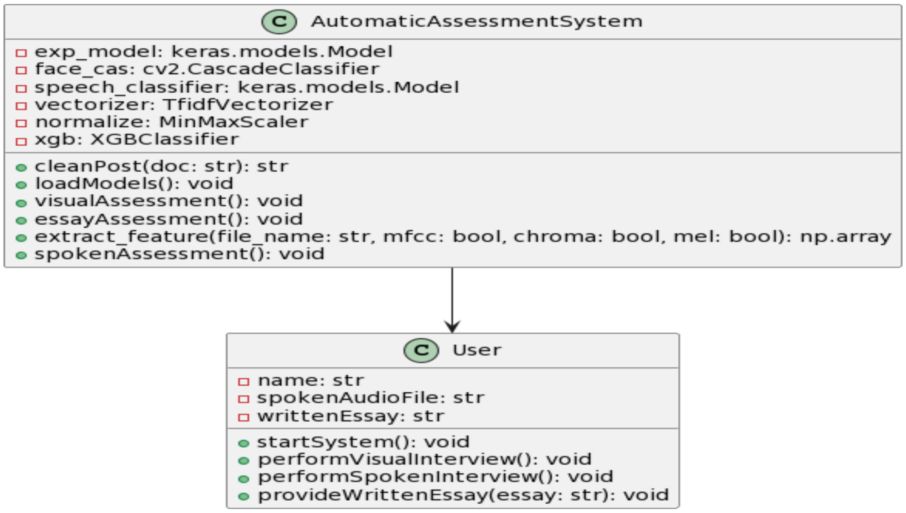

C AutomaticAssessmentSystem
exp_model: keras.models.Model
face_cas: cv2.CascadeClassifier
speech_classifier: keras.models.Model
vectorizer: TfidfVectorizer
normalize: MinMaxScaler
xgb: XGBClassifier
cleanPost(doc: str): str
loadModels(): void
visualAssessment(): void
essayAssessment(): void
extract_feature(file_name: str, mfcc: bool, chroma: bool, mel: bool): np.array
spokenAssessment(): void
C User
name: str
spokenAudioFile: str
writtenEssay: str
startSystem(): void
performVisualInterview(): void
performSpokenInterview(): void
provideWrittenEssay(essay: str): void

Fig: 5.2.1 Class diagram

Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

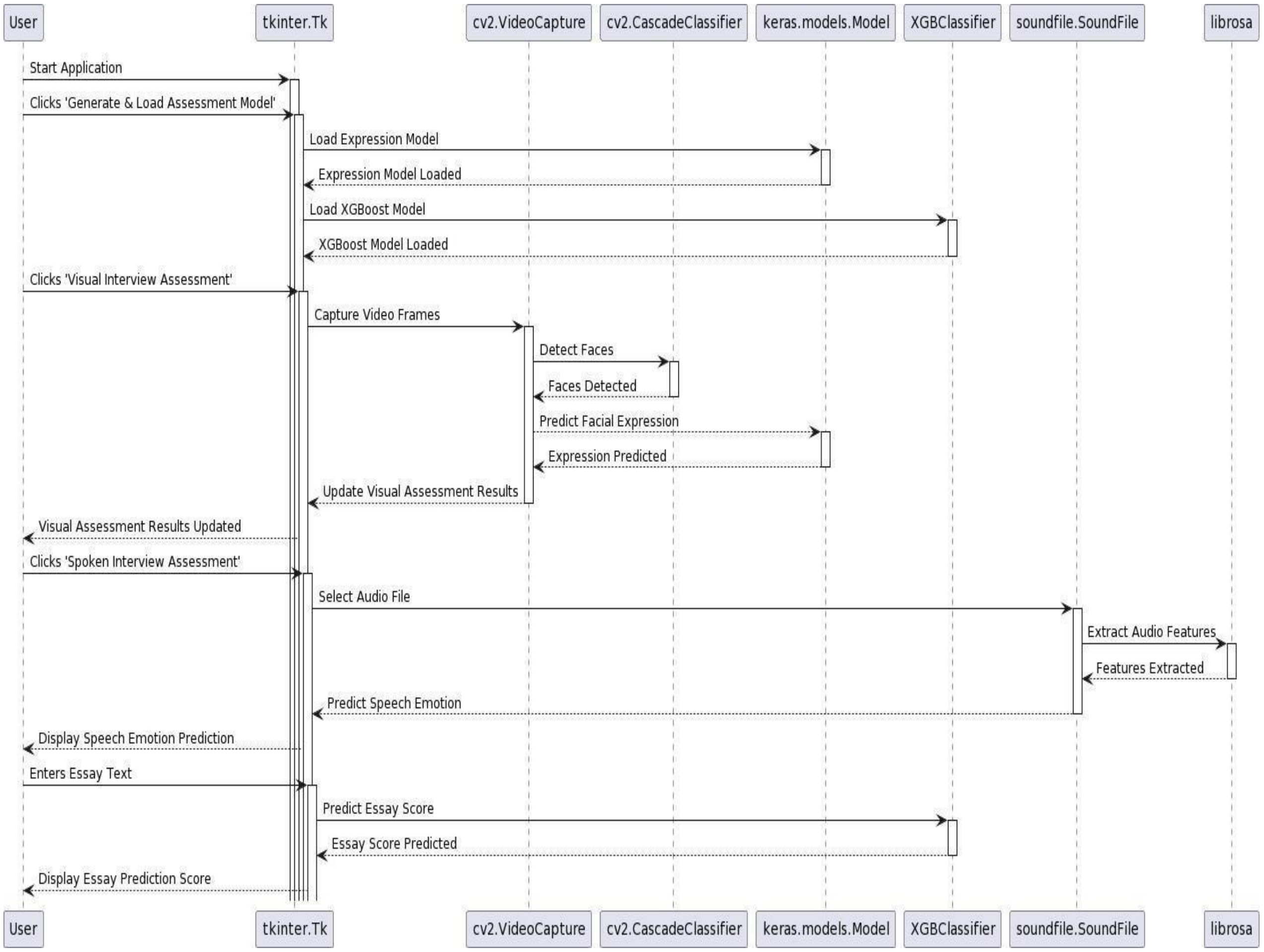


Fig: 5.2.2 Sequence diagram

Use case diagram

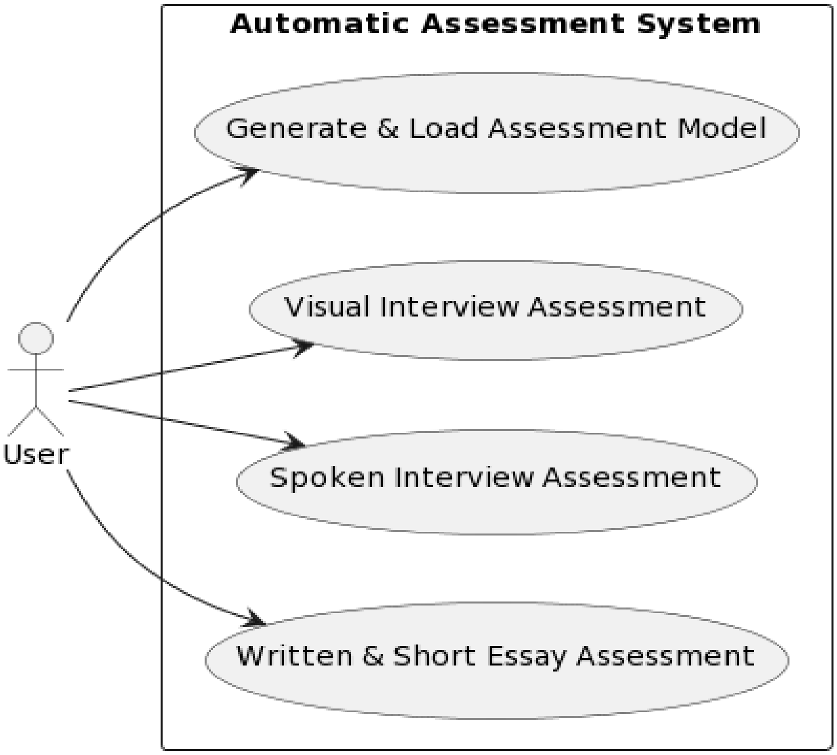

Automatic Assessment System
Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment
User

Fig: 5.2.3 Use case diagram

Activity diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

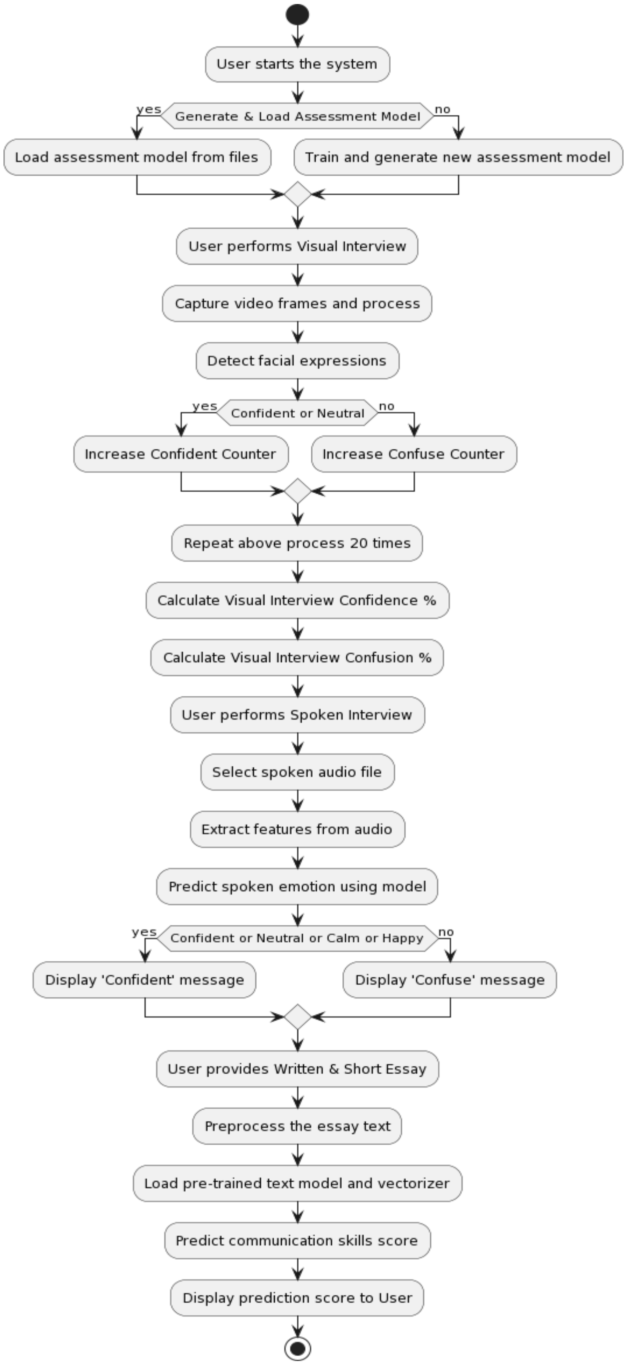

User starts the system
yes
Generate & Load Assessment Model
no
Load assessment model from files
Train and generate new assessment model
User performs Visual Interview
Capture video frames and process
Detect facial expressions
yes
Confident or Neutral
no
Increase Confident Counter
Increase Confuse Counter
Repeat above process 20 times
Calculate Visual Interview Confidence %
Calculate Visual Interview Confusion %
User performs Spoken Interview
Select spoken audio file
Extract features from audio
Predict spoken emotion using model
yes
Confident or Neutral or Calm or Happy
no
Display 'Confident' message
Display 'Confuse' message
User provides Written & Short Essay
Preprocess the essay text
Load pre-trained text model and vectorizer
Predict communication skills score
Display prediction score to User

Fig: 5.2.4 Activity diagram

6.CODING AND IMPLEMENTATION

6.1Sample Code

```
from tkinter import *
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
from PIL import Image
import tensorflow as tf
from tensorflow import keras
import numpy as np
import cv2
import os
from tkinter import messagebox
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns
from sklearn.model_selection import train_test_split
from string import punctuation
from nltk.corpus import stopwords
import nltk
```

```
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
import pickle
from nltk.stem import PorterStemmer
from sklearn.preprocessing import MinMaxScaler
from xgboost import XGBClassifier
import soundfile
import librosa
from keras.models import model_from_json
main = tkinter.Tk()
main.title("Active Online Learning for Social media Analysis to Support Crisis Management")
main.geometry("1200x1200")
facial_expression= ['Anger', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']
speech_emotion = ['neutral', 'calm', 'happy', 'sad', 'angry', 'fearful', 'disgust', 'surprised']
exp_model = keras.models.load_model("model/model_35_91_61.h5")
font_cv = cv2.FONT_HERSHEY_SIMPLEX
face_cas = cv2.CascadeClassifier('model/haarcascade_frontalface_default.xml')
global video, vectorizer, normalize, xgb
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()
ps = PorterStemmer()
with open('model/speechmodel.json', "r") as json_file:
loaded_model_json = json_file.read()
speech_classifier = model_from_json(loaded_model_json)
json_file.close()
```

```
speech_classifier.load_weights("model/speech_weights.h5")
speech_classifier._make_predict_function()
def cleanPost(doc):
    tokens = doc.split()
    table = str.maketrans('', '', punctuation)
    tokens = [w.translate(table) for w in tokens]
    tokens = [word for word in tokens if word.isalpha()]
    tokens = [w for w in tokens if not w instop_words]
    tokens = [word for word in tokens if len(word) > 1]
    tokens = [ps.stem(token) for token in tokens]
    tokens = [lemmatizer.lemmatize(token) for token in tokens]
    tokens = ' '.join(tokens)
    return tokens
def loadModels():
    global vectorizer, normalize, xgb
text.delete('1.0', END)
textdata = np.load("model/X.npy")
    Y = np.load("model/Y.npy")
    vectorizer = TfidfVectorizer(stop_words=stop_words, use_idf=True, smooth_idf=False, norm=None, decode_error='replace', max_features=3000)
    X = vectorizer.fit_transform(textdata).toarray()
    indices = np.arange(X.shape[0])
np.random.shuffle(indices)
    X = X[indices]
    Y = Y[indices]
    normalize = MinMaxScaler()
    X = normalize.fit_transform(X)
    print(X.shape)
    print(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
    if os.path.exists("model/xgb.txt"):
        with open('model/xgb.txt', 'rb') as file:
xgb = pickle.load(file)
file.close()
    else:
xgb = XGBClassifier()
xgb.fit(X_train, y_train)
        with open('model/xgb.txt', 'wb') as file:
pickle.dump(xgb, file)
file.close()
    predict = xgb.predict(X_test)
    p = precision_score(y_test, predict,average='macro') * 100
    r = recall_score(y_test, predict,average='macro') * 100
    f = f1_score(y_test, predict,average='macro') * 100
    a = accuracy_score(y_test,predict)*100
text.insert(END,"XGBoost Accuracy : "+str(a)+"\n")
text.insert(END,"XGBoost Precision : "+str(p)+"\n")
text.insert(END,"XGBoost Recall    : "+str(r)+"\n")
text.insert(END,"XGBoost FSCORE    : "+str(f)+"\n\n")
    labels = np.unique(y_test)
conf_matrix = confusion_matrix(y_test, predict)
plt.figure(figsize =(6, 6))
```

```
ax = sns.heatmap(conf_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis"
,fmt ="g");

ax.set_ylim([0,len(labels)])

plt.title("XGBoost Communication SKills Score Prediction Confusion Matrix Graph")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

def visualAssessment():

text.delete('1.0', END)

    counter = 0

    confident = 0

    confuse = 0

    video = cv2.VideoCapture(0)

while(counter < 20):

        ret, frame = video.read()

        if ret == True:

            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

            faces = face_cas.detectMultiScale(gray, 1.3,5)

            for   (x,   y,   w,   h)   in   faces:

face_component = gray[y:y+h, x:x+w]

                fc = cv2.resize(face_component, (48, 48))

inp  =  np.reshape(fc,(1,48,48,1)).astype(np.float32)

inp = inp/255.

                prediction = exp_model.predict_proba(inp)

                expression = facial_expression[np.argmax(prediction)]

                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
                if expression == 'Neutral' or expression == 'Happy':
                    confident = confident + 1
                else:
                    confuse = confuse + 1
                counter = counter + 1
                print(counter)
                cv2.putText(frame, "Confident Count : "+str(confident), (30, 40), font_cv, 1, (0, 255, 0),2)
cv2.putText(frame, "Confuse Count : "+str(confuse), (30, 120), font_cv, 1, (0, 255, 0), 2)
            cv2.imshow("image", frame)
            if cv2.waitKey(250) & 0xFF == ord('q'):
                break
        else:
            break
video.release()
    cv2.destroyAllWindows()
    if confident > 0:
        confident = confident / 20.0
    if confuse > 0:
        confuse = confuse / 20.0
text.insert(END,"Your Visual Interview Confidence% : "+str(confident)+"\n")
text.insert(END,"Your Visual Interview Confusion% : "+str(confuse)+"\n")
text.update_idletasks()
def essayAssessment():
    global vectorizer, normalize, xgb
    essay = text.get(1.0, "end-1c")
    print(essay)
    state = essay.strip().lower()
    state = cleanPost(state)
    temp = []
temp.append(state)
    temp = vectorizer.transform(temp).toarray()
    temp = normalize.transform(temp)
    predict = xgb.predict(temp)
    predict = predict[0]
messagebox.showinfo("Your Essay Prediction Score : "+str(predict), "Your Essay Prediction Score : "+str(predict))
def extract_feature(file_name, mfcc, chroma, mel):
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
sample_rate=sound_file.samplerate
        if chroma:
stft=np.abs(librosa.stft(X))
        result=np.array([])
        if mfcc:
mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
            result=np.hstack((result, mfccs))
        if chroma:
            chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
            result=np.hstack((result, chroma))
        if mel:
mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
```

```
sound_file.close()
    return result
def spokenAssessment():
    global speech_classifier
    filename = filedialog.askopenfilename(initialdir="testSpeech")
fname = os.path.basename(filename)
    test = []
mfcc = extract_feature(filename, mfcc=True, chroma=True, mel=True)
test.append(mfcc)
    test = np.asarray(test)
    test = test.astype('float32')
    test = test/255
test = test.reshape((test.shape[0],test.shape[1],1,1))
    predict = speech_classifier.predict(test)
    predict = np.argmax(predict)
    predict = speech_emotion[predict-1]
    if predict == 'neutral' or predict == 'calm' or predict == 'happy':
messagebox.showinfo("Your Speaking Verbal Audio Predicted as : Confident","Your Speaking Verbal Audio Predicted as : Confident")
    else:
messagebox.showinfo("Your Speaking Verbal Audio Predicted as : Confuse","Your Speaking Verbal Audio Predicted as : Confuse")
font = ('times', 14, 'bold')
title = Label(main, text='Active Online Learning for Social media Analysis to Support Crisis Management')
title.config(bg='DarkGoldenrod1', fg='black')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=5,y=5)
font1 = ('times', 13, 'bold')
loadButton = Button(main, text="Generate & Load Assessment Model", command=loadModels)
loadButton.place(x=50,y=250)
loadButton.config(font=font1)
videoButton = Button(main, text="Visual Interview Assessment", command=visualAssessment)
videoButton.place(x=400,y=250)
videoButton.config(font=font1)
spokenButton = Button(main, text="Spoken Interview Assessment", command=spokenAssessment)
spokenButton.place(x=690,y=250)
spokenButton.config(font=font1)
essayButton = Button(main, text="Written & Short Essay Assessment", command=essayAssessment)
essayButton.place(x=990,y=250)
essayButton.config(font=font1)
font1 = ('times', 12, 'bold')
text=Text(main,height=13,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=400)
text.config(font=font1)
main.config(bg='LightSteelBlue1')
main.mainloop()
```

6.2 Data Dictionary

This project implements an interactive application using the Tkinter library. The application is designed for active online learning for social media analysis to support crisis management. Let's break down the functionality and structure of the code:

- **Imports**: The script begins with necessary imports. It imports various modules and libraries such as Tkinter for GUI, PIL for image processing, TensorFlow and Keras for deep learning, OpenCV for computer vision, and others for text processing, machine learning, and audio analysis.
- **Global Variables**: Several global variables are declared to hold models, classifiers, fonts, and lists related to facial expressions and speech emotions.
- **Tkinter Setup**: The script initializes a Tkinter window (**main**) with a title and geometry. The title reflects the purpose of the application, which is "Active Online Learning for Social media Analysis to Support Crisis Management."
- **Functions**:
    - **Clean Post**: This function preprocesses text data by tokenizing, removing punctuation, stopwords, and applying stemming and lemmatization.
    - **Load Models**: Loads and trains machine learning models for text classification using XGBoost. It preprocesses the data, splits it into training and testing sets, and evaluates the model's performance metrics such as accuracy, precision, recall, and F1-score. It also displays a confusion matrix.
    - **Visual Assessment**: Performs real-time facial expression analysis using a webcam. It detects faces, extracts facial components, predicts expressions using a pre-trained deep learning model, and calculates confidence and confusion percentages based on the detected expressions.
    - **Essay Assessment**: Analyzes written essays provided by the user. It preprocesses the text, vectorizes it using TF-IDF, normalizes it, and predicts the score using a pre-trained XGBoost model.

  - **extract_feature**: Extracts features from audio files for spoken interview assessment. It computes Mel-frequency cepstral coefficients (MFCC), chroma feature, and mel-scaled spectrogram features.
  - **Spoken Assessment**: Performs spoken interview assessment by analyzing audio files. It extracts features, preprocesses the data, and predicts speech emotions using a pre-trained deep learning model.

- **GUI Elements**:
  - **Labels**: A title label and various buttons for model loading, visual interview assessment, spoken interview assessment, and written essay assessment.
  - **Text Widget**: A text widget to display output messages, performance metrics, and analysis results.
- **GUI Configuration**:
  - The appearance and positioning of GUI elements are configured using **place** and **config** methods. The buttons are configured with appropriate command functions for event handling.
- **Execution**: The Tkinter window is configured with background color, and the main event loop (**mainloop()**) is initiated to run the application.

7.SYSTEM TESTING

7.1Testing Strategies

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Testing

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies onknowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and containsclearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. You cannot “see” into it. The test provides inputs and responds to outputs.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or moreintegrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. componentsin a software system or – one step up – software applications at the company level – interact withouterror.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requiressignificant participation. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

8.OUTPUT SCREENS

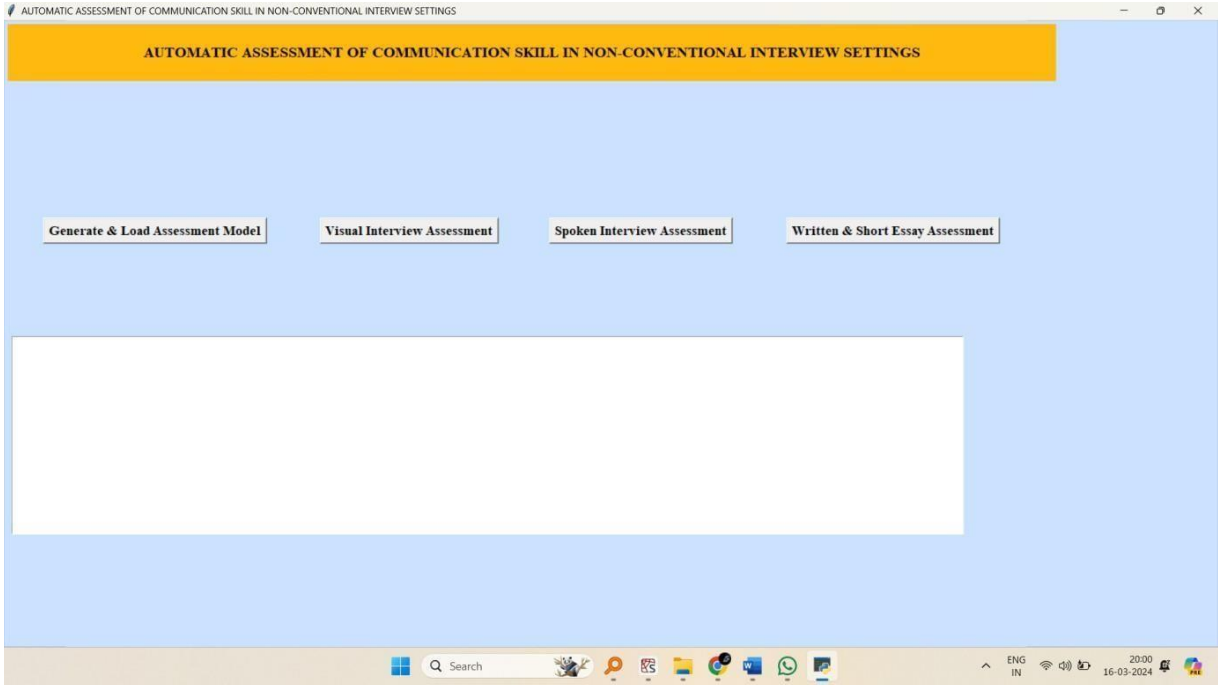
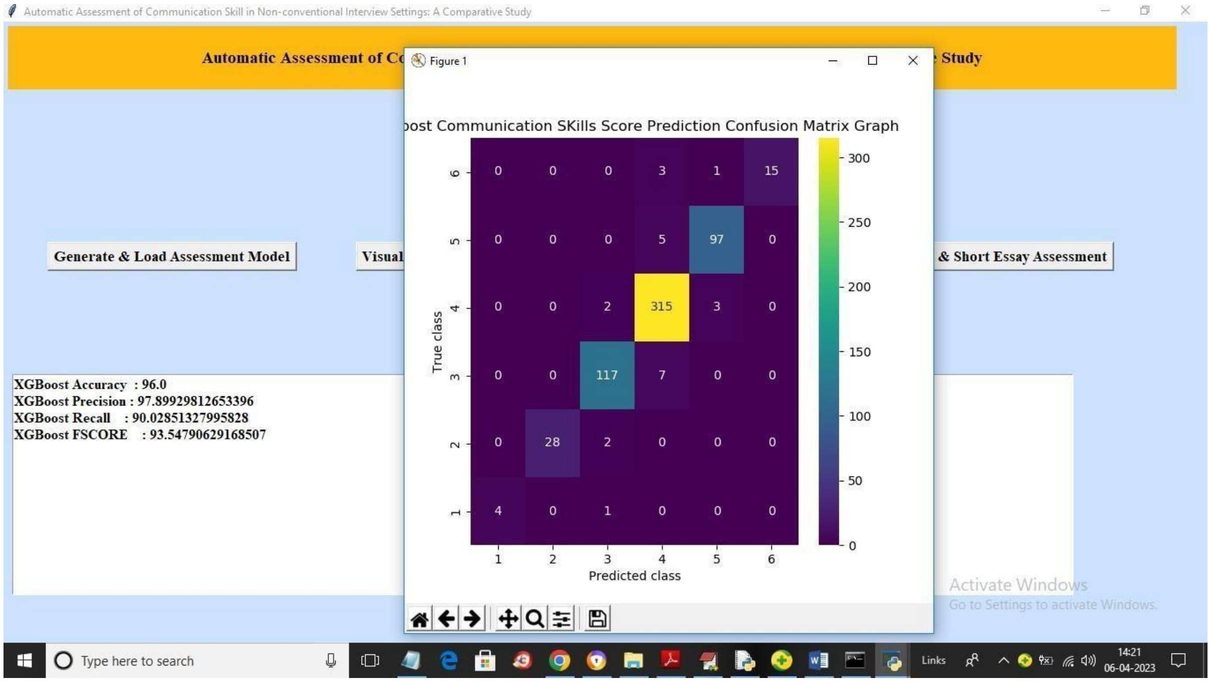

AUTOMATIC ASSESSMENT OF COMMUNICATION SKILL IN NON-CONVENTIONAL INTERVIEW SETTINGS
AUTOMATIC ASSESSMENT OF COMMUNICATION SKILL IN NON-CONVENTIONAL INTERVIEW SETTINGS
Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment
Search
ENG
IN
20:00
16-03-2024

Figure 1: GUI interface of Interview Settings.

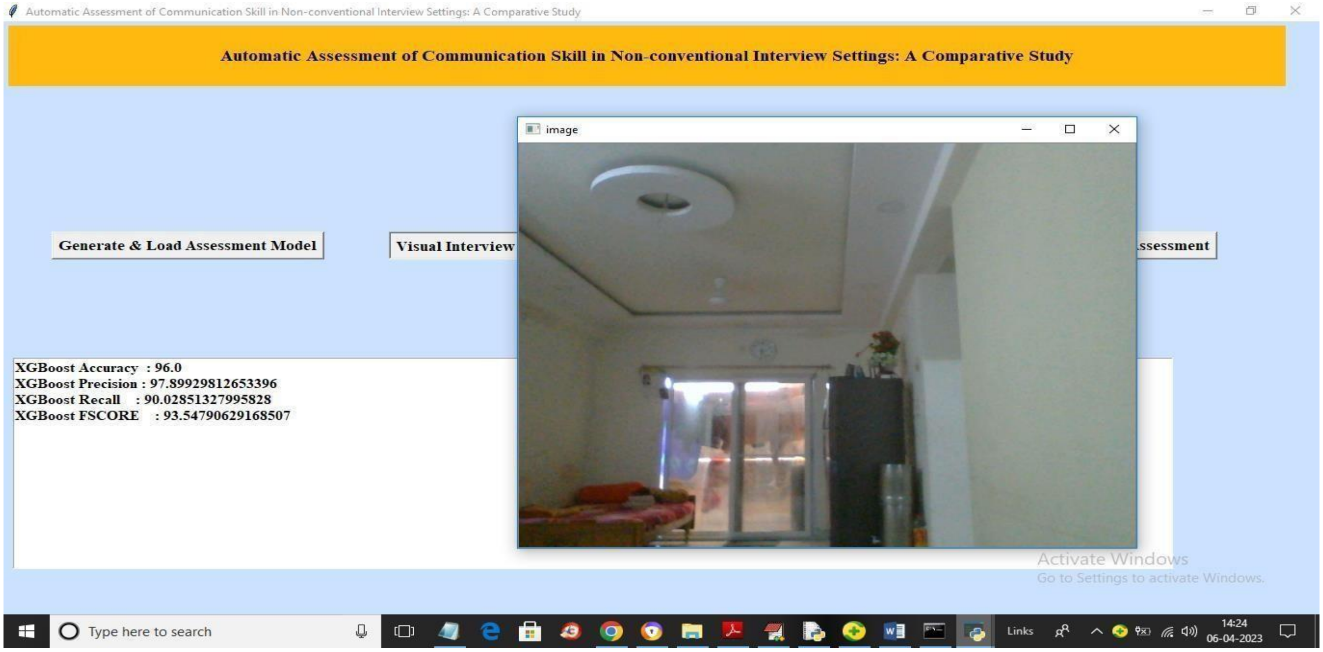

Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
Automatic Assessment of Co
Study
Generate & Load Assessment Model
Visual
& Short Essay Assessment
Figure 1
ost Communication SKills Score Prediction Confusion Matrix Graph
True class
Predicted class
300
250
200
150
100
50
0
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:21
06-04-2023

True class	1	2	3	4	5	6
6	0	0	0	3	1	15
5	0	0	0	5	97	0
4	0	0	2	315	3	0
3	0	0	117	7	0	0
2	0	28	2	0	0	0
1	4	0	1	0	0	0

```
XGBoost Accuracy  : 96.0
XGBoost Precision : 97.89929812653396
XGBoost Recall    : 90.02851327995828
XGBoost FSCORE    : 93.54790629168507
```

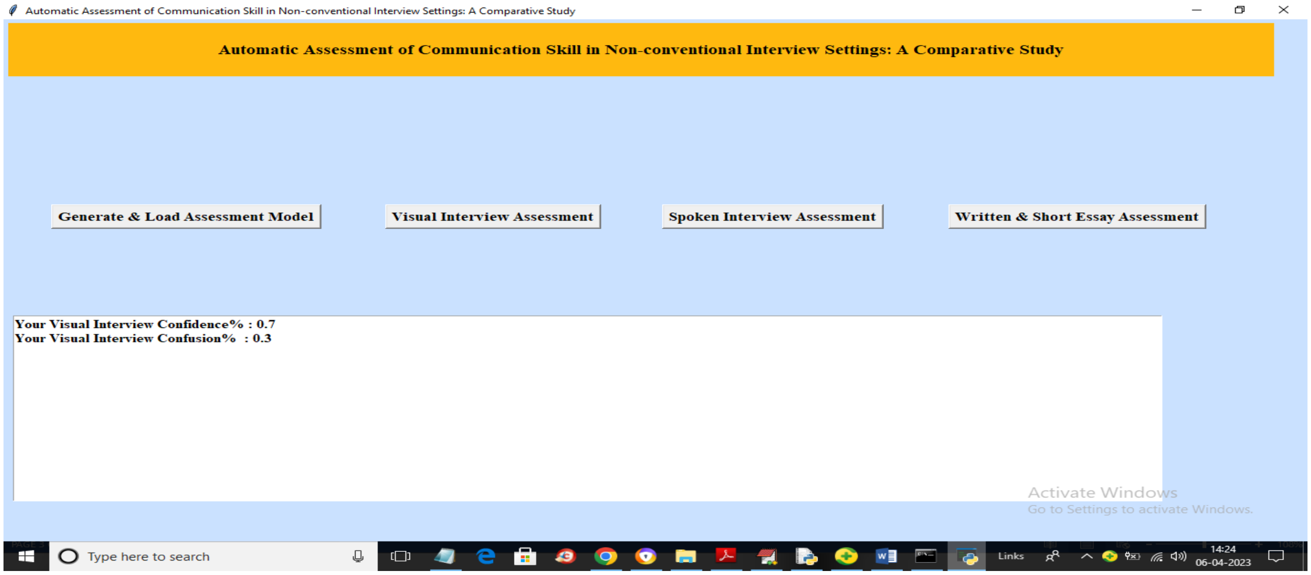
Figure 2: Displays the Accuracy of XGBoost model and its confusion matrix.

Figure 2 Presents the models loaded and with XGBOOST we got 96% accuracy and we can see other metrics also and in confusion matrix graph x-axis represents predicted ESSAY score from 1 to 6 and y-axis represents TRUE Score LABELS and all different color boxes represents correct prediction count and all blue boxes represents incorrect prediction count. Now click on 'Visual Interview Assessment' button to start Webcam and monitor person for 20 frames with faces and then give average score


Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
image
Generate & Load Assessment Model
Visual Interview
ssessment
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:24
06-04-2023

```
XGBoost Accuracy  : 96.0
XGBoost Precision : 97.89929812653396
XGBoost Recall    : 90.02851327995828
XGBoost FSCORE    : 93.54790629168507
```

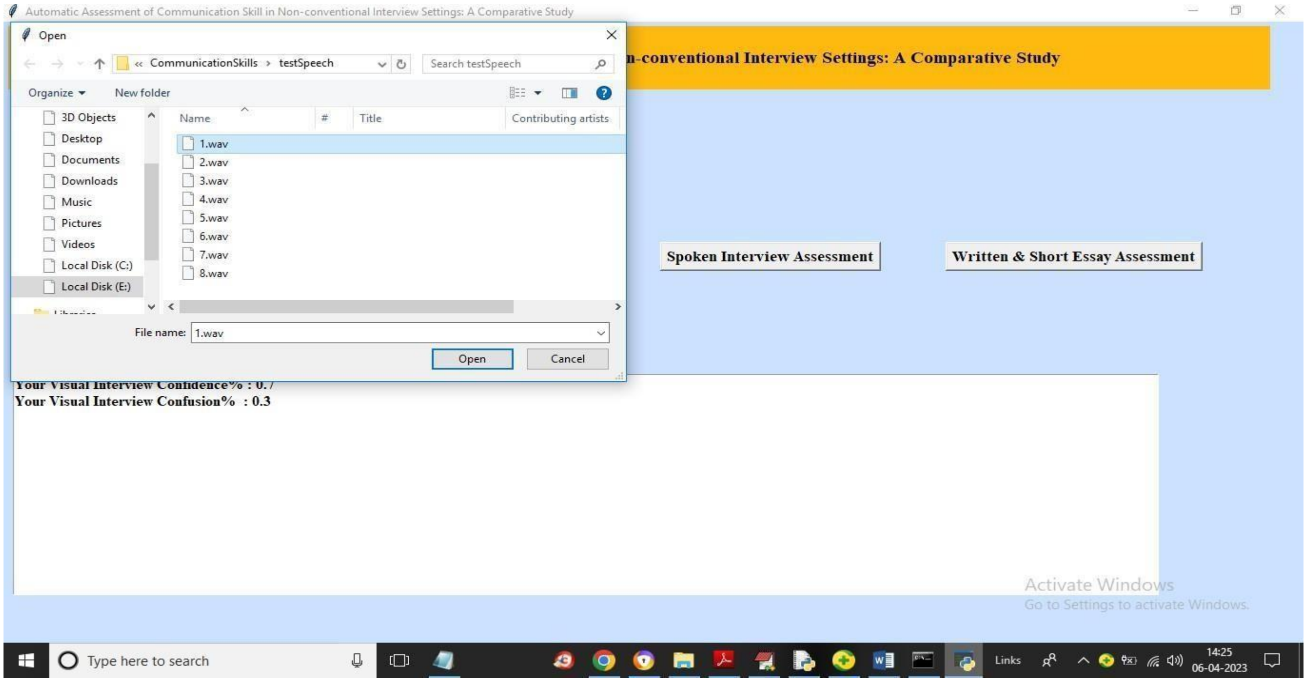
Figure 3: Presents the webcam for candidate confidence detection.


Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:24
06-04-2023

```
Your Visual Interview Confidence% : 0.7
Your Visual Interview Confusion%  : 0.3
```

Figure 4: Presents the confidence level of candidate

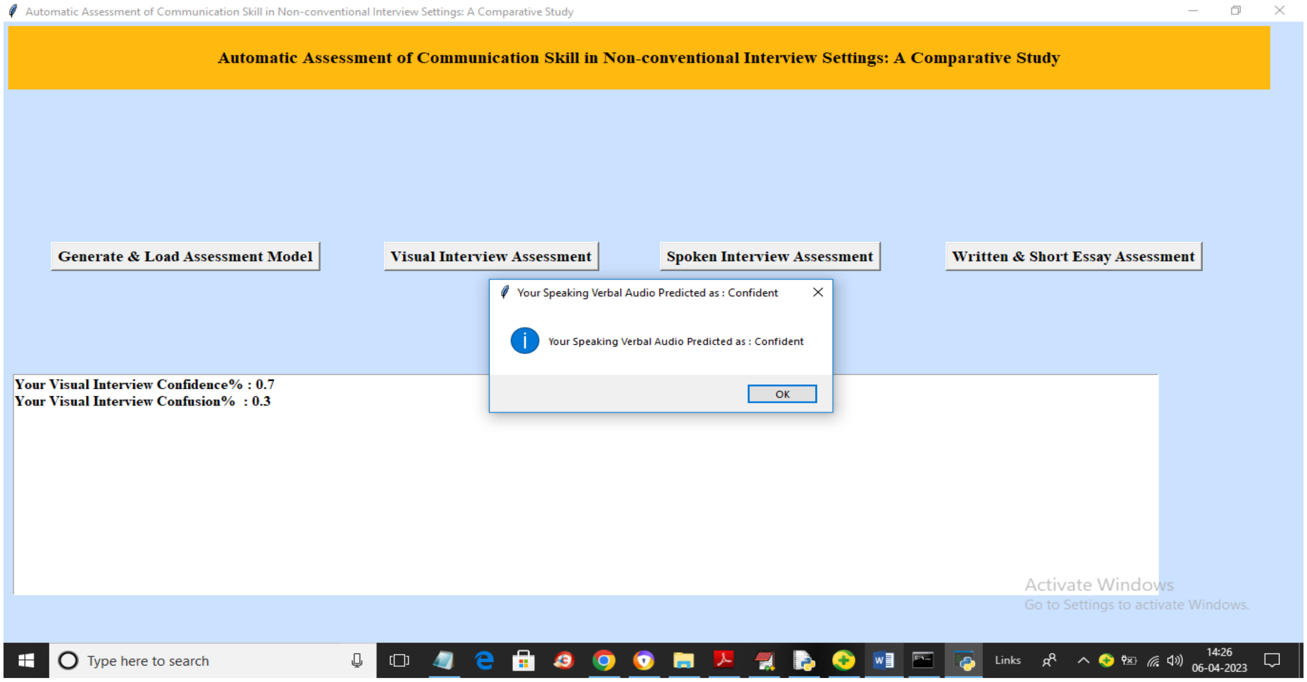
Figure 4 Got average confident and confusion score and now click on 'Spoken Interview Assessment' button to upload audio file and then predict person skills based on audio voice features


Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
n-conventional Interview Settings: A Comparative Study
Open
« CommunicationSkills › testSpeech
Search testSpeech
Organize
New folder
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Name
#
Title
Contributing artists
1.wav
2.wav
3.wav
4.wav
5.wav
6.wav
7.wav
8.wav
Open
Cancel
Spoken Interview Assessment
Written & Short Essay Assessment
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:25
06-04-2023

File name: 1.wav

```
Your Visual Interview Confidence% : 0.7
Your Visual Interview Confusion%  : 0.3
```

Figure 5: Presents the selecting and uploading audio file.



Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study

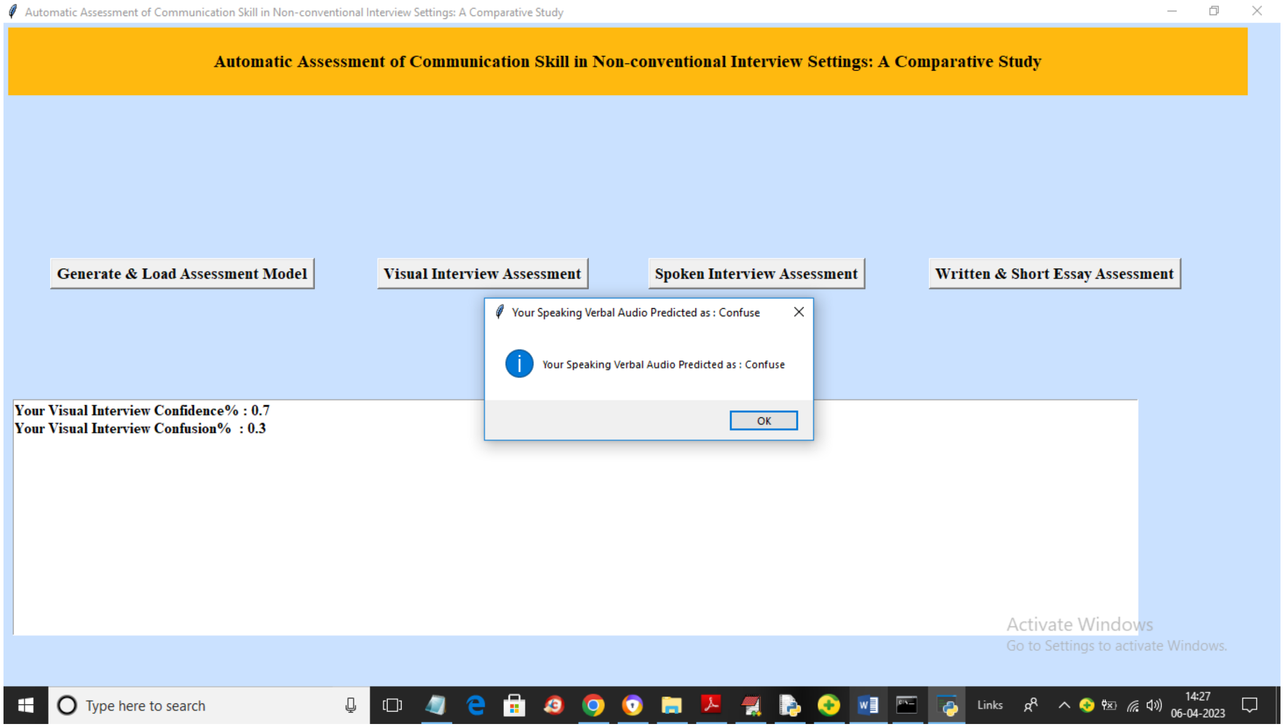
Your Speaking Verbal Audio Predicted as : Confident

Your Speaking Verbal Audio Predicted as : Confident

```
Your Visual Interview Confidence% : 0.7
Your Visual Interview Confusion%  : 0.3
```

Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment
OK
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:26
06-04-2023

Figure 6: Displays the Dialog box we got output as ‘Confident’.



Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study

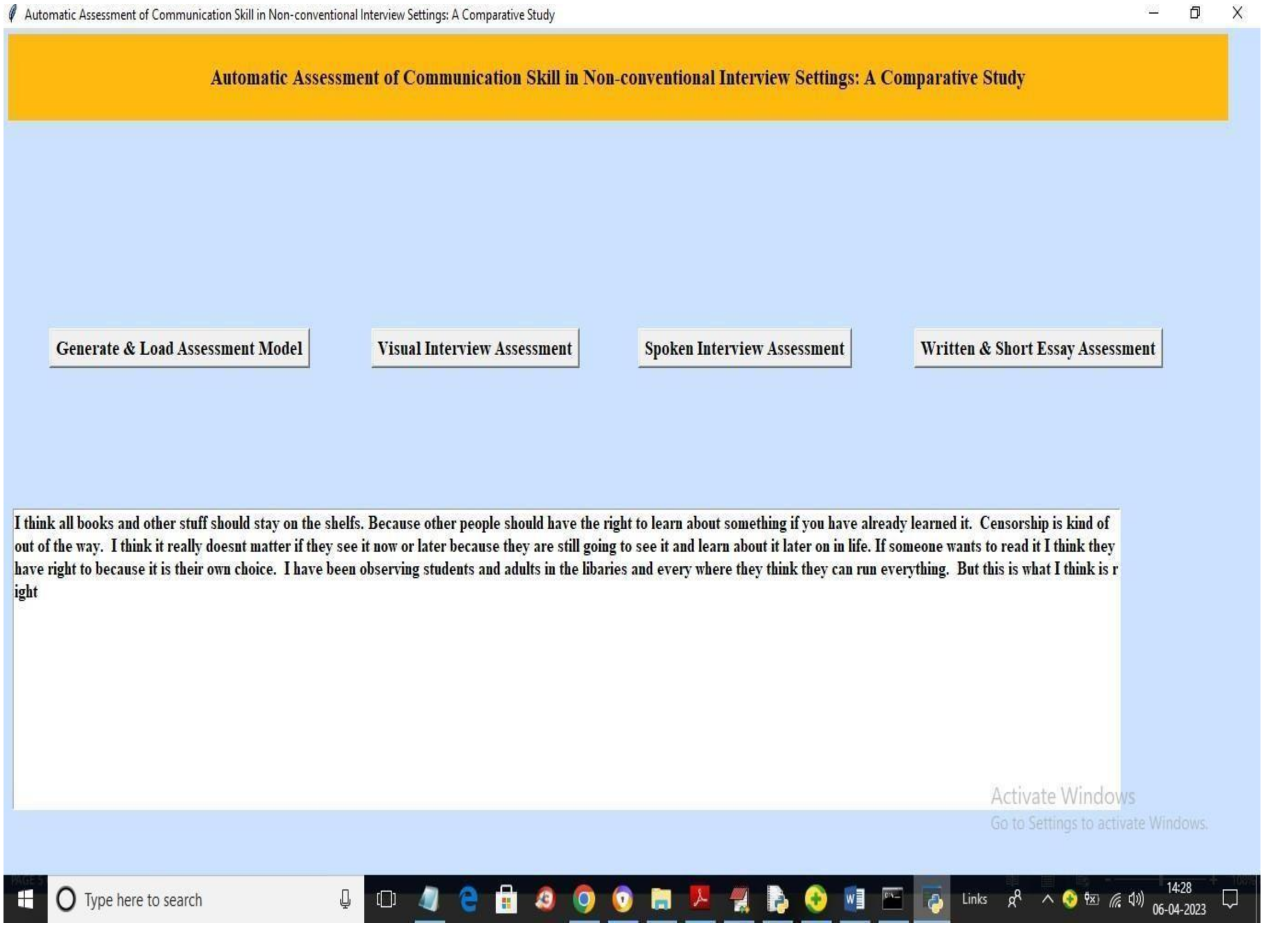
Your Speaking Verbal Audio Predicted as : Confuse

Your Speaking Verbal Audio Predicted as : Confuse

```
Your Visual Interview Confidence% : 0.7
Your Visual Interview Confusion%  : 0.3
```

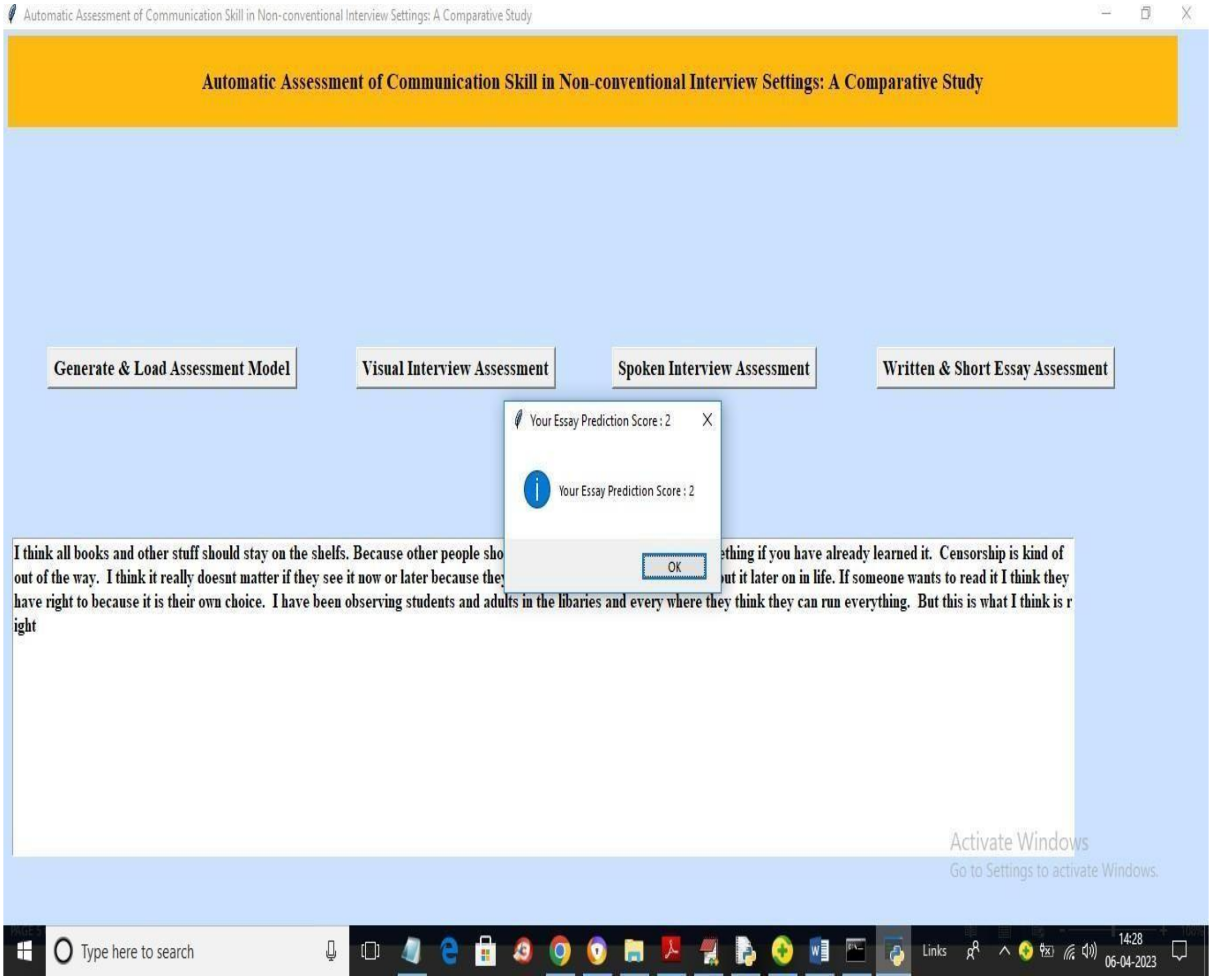
Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment
OK
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:27
06-04-2023

Figure 7: Displays the Dialog box we got output as ‘Confused’.


Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study
Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment
Activate Windows
Go to Settings to activate Windows.
Type here to search
Links
14:28
06-04-2023

I think all books and other stuff should stay on the shelfs. Because other people should have the right to learn about something if you have already learned it. Censorship is kind of
out of the way. I think it really doesnt matter if they see it now or later because they are still going to see it and learn about it later on in life. If someone wants to read it I think they
have right to because it is their own choice. I have been observing students and adults in the libaries and every where they think they can run everything. But this is what I think is r
ight

Figure 8: The text area is written some essay.



Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study

Automatic Assessment of Communication Skill in Non-conventional Interview Settings: A Comparative Study

Generate & Load Assessment Model
Visual Interview Assessment
Spoken Interview Assessment
Written & Short Essay Assessment

Your Essay Prediction Score : 2

Your Essay Prediction Score : 2

I think all books and other stuff should stay on the shelfs. Because other people sho
ething if you have already learned it. Censorship is kind of
out of the way. I think it really doesnt matter if they see it now or later because they
ut it later on in life. If someone wants to read it I think they
have right to because it is their own choice. I have been observing students and adults in the libaries and every where they think they can run everything. But this is what I think is r
ight

Figure 9: Presents the essay score predicted as ‘2’.

9.CONCLUSION

In the rapidly evolving landscape of recruitment, the need for effective and objective assessment methods for communication skills has never been more crucial. The proposed project, focusing on automatic assessment of communication skills in non-conventional interview settings, represents a significant step towards revolutionizing the way candidates are evaluated in the modern job market. By harnessing the power of technology, including machine learning algorithms, natural language processing, and computer vision, this project offers a comprehensive solution to the limitations of traditional communication skills assessment methods. The system provides several key advantages, including objective evaluations, efficiency, cost-effectiveness, and adaptability, making it a valuable asset for both employers and job applicants. Through visual assessments, spoken evaluations, and essay analysis, candidates are given fair and data-driven feedback, enabling them to understand their strengths and areas for improvement. Employers benefit from streamlined, consistent, and scalable assessments, leading to more informed hiring decisions and a diverse, skilled workforce. The project's success lies in its ability to bridge the gap between traditional interview methods and the demands of the digital age. By offering remote accessibility, unbiased evaluations, and insightful feedback, the system ensures a seamless and inclusive recruitment process for candidates from various backgrounds and locations.

10.FUTURE ENHANCEMENTS

In the future, the automatic assessment of communication skills in non-conventional interview settings is set to revolutionize with innovations such as advanced NLP for deeper understanding of language nuances, AR and VR to simulate real-world interaction scenarios, and AI-driven behavioral analysis for comprehensive non-verbal cues evaluation. These technologies will offer interactive, gamified environments for a more engaging assessment experience while ensuring ethical AI use and continuous learning systems adapt to evolving communication standards. Together, these enhancements will enable a more holistic, customizable, and accurate evaluation of candidates' communication skills, ensuring they are well-suited for specific roles and organizational cultures, while maintaining the essential human element in the decision-making process. The continuous improvement and refinement of this automated assessment system will be essential. Feedback from users and the analysis of assessment data will guide future enhancements, ensuring the system remains aligned with the dynamic requirements of the job market.

11.REFERENCES

[1] 2016. Discover Big Voice Intelligence. https://www.voicebase.com/ products-features/. (2016).

[2] Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® V. 2. The Journal of Technology, Learning and Assessment 4, 3 (2006).

[3] Ligia Maria Batrinca, Nadia Mana, Bruno Lepri, Fabio Pianesi, and Nicu Sebe. 2011. Please, tell me about yourself: automatic personality assessment using short self-presentations. In Proceedings of the 13th international conference on multimodal interfaces. ACM, 255–262.

[4] Joan-Isaac Biel, Lucía Teijeiro-Mosquera, and Daniel Gatica-Perez. 2012. Facetube: predicting personality from facial expressions of emotion in online conversational video. In Proceedings of the 14th ACM international conference on Multimodal interaction. ACM, 53–56.

[5] Laurent Son Nguyen, Denise Frauendorfer, Marianne Schmid Mast, and Daniel Gatica-Perez. 2014. Hire me: Computational inference of hirability in employment interviews based on nonverbal behavior. IEEE transactions on multimedia 16, 4 (2014), 1018–1031.

[6] Sowmya Rasipuram and Dinesh Babu Jayagopi. 2016. Automatic assessment of communication skill in interface-based employment interviews using audiovisual cues. In Multimedia & Expo Workshops (ICMEW), 2016 IEEE International Conference on. IEEE, 1–6.

[7] Iftekhar Naim, M Iftekhar Tanveer, Daniel Gildea, and Mohammed Ehsan Hoque. 2015. Automated prediction and analysis of job interview performance: The role of what you say and how you say it. In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, Vol. 1. IEEE, 1–6.

[8] Sowmya Rasipuram, Dinesh Babu Jayagopi, et al. 2016. Asynchronous video interviews vs. face-to-face interviews for communication skill measurement: a systematic study. In Proceedings of the 18th ACM International Conference on Multimodal Interaction. ACM, 370–377.

[9] Mohammed Ehsan Hoque, Matthieu Courgeon, Jean-Claude Martin, Bilge Mutlu, and Rosalind W Picard. 2015. Mach: My automated conversation coach. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 385–396.

[10] M Iftekhar Tanveer, Emy Lin, and Mohammed Ehsan Hoque. 2015. Rhema: A real-time in-situ intelligent interface to help people with public speaking. In Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, 286–295.

[11] Michelle Fung, Yina Jin, RuJie Zhao, and Mohammed Ehsan Hoque. 2015. ROC speak: semi-automated personalized feedback on nonverbal behavior from recorded videos. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 1167–1178.

[12] M Iftekhar Tanveer, Ru Zhao, Kezhen Chen, Zoe Tiet, and Mohammed Ehsan Hoque. 2016. Automanner: An automated interface for making public speakers aware of their mannerisms. In Proceedings of the 21st International Conference on Intelligent User Interfaces. ACM, 385–396.

[13] Hiroki Tanaka, Sakriani Sakti, Graham Neubig, Tomoki Toda, Hideki Negoro, Hidemi Iwasaka, and Satoshi Nakamura. 2013. Automated social skills trainer. In Proceedings of the 20th International Conference on Intelligent User Interfaces. ACM, 697–706.

[14] Brian H Spitzberg and Thomas W Adams. 2007. CSRS, the conversational skills rating scale: an instructional assessment of interpersonal competence. NCA, National Communication Association.

[15] Ellis B Page. 1966. The imminence of... grading essays by computer. The Phi Delta Kappan 47, 5 (1966), 238–243.

[16] Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education 25, 1 (2015), 60–117.

[17] Jana Z Sukkarieh and Stephen G Pulman. 2005. Information extraction and machine learning: Auto-marking short free text responses to science questions. In Proceedings of the 2005 conference on artificial intelligence in education: Supporting learning through intelligent and socially informed technology. IOS Press, 629–637.

[18] Louis Tandalla. 2012. Scoring Short Answer Essays. (2012). https://kaggle2.blob.core.windows.net/competitions/kaggle/2959/media/TechnicalMethodsPaper.pdf.

[19] Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications. 97–106.

[20] Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 752–762.

[21] Detmar Meurers, Ramon Ziai, Niels Ott, and Janina Kopp. 2011. Evaluating answers to reading comprehension questions in context: Results for German and the role of information structure. In Proceedings of the TextInfer 2011 Workshop on Textual Entailment. Association for Computational Linguistics, 1–9.

[22] Derrick Higgins, Chris Brew, Michael Heilman, Ramon Ziai, Lei Chen, Aoife Cahill, Michael Flor, Nitin Madnani, Joel Tetreault, Daniel Blanchard, et al. 2014. Is getting the right answer just about choosing the right words? The role of syntactically-informed features in short answer scoring. arXiv preprint arXiv:1403.0801 (2014).

Text Books:

1.Python: The Complete Reference, Martin C. Brown, Mc Graw Hill Education; Forth edition.

2 .Machine Learning, Tom M. Mitchell, McGraw Hill Education; First edition.

WEBSITES:

1. https://www.w3schools.com/python/
2. https://www.geeksforgeeks.org/machine-learning/
3. https://www.journals.elsevier.com/speech-communication