AUTOMATED PAPER EVALUATION

A Major project Report submitted to

Rajiv Gandhi University of Knowledge and Technologies,
SRIKAKULAM

in partial fulfilment of the requirements for the
Award of the degree of

Bachelor of Technology in
Computer Science and Engineering

Submitted by

K. Priyanka	**S180124**
Ch. Geetha	**S180402**
Sk. Samdan	**S180716**
T. Satya Priya	**S180130**

Under the Supervision of

Mr.T.Anil Kumar
Asst. Professor – Department of CSE

Department of Computer Science and Engineering
Rajiv Gandhi University of Knowledge Technologies
SRIKAKULAM

CERTIFICATE

This is to certify that the report entitled “**Automated Paper Evaluation**” was submitted by **Priyanka Kommani**, bearing ID. No. S180124, **Geetha Chikati**, bearing ID. No. S180402, **Samdan Shaik**, bearing ID. NO S180716, **Satya Priya Telukula**, bearing ID. No. S180130 in partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science is a bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution to award any degree or diploma

Mr.T.Anil Kumar,
Asst.prof.,Dept.of CSE
RGUKT SRIKAKULAM.

Mrs.Ch.Lakshmibala
Head of the Department,
RGUKT SRIKAKULAM.

DECLARATION

We**, Priyanka Kommani, Geetha Chikati, Samdan Shaik, Satya Priya Telukula** hereby declare that this report entitled "**Automated Paper Evaluation**" submitted by us under the guidance and supervision of Mr.**T.Anil Kumar** is a bonafide work.

We also declare that it has not been submitted previously in part or in full to this University or other University or Institution to award any degree or diploma.

Priyanka Kommani	**S180124**
Geetha Chikati	**S180402**
Samdan Shaik	**S180716**
Satya Priya Telukula	**S180130**

Acknowledgements

We would like to express my sincere gratitude to, my project Guide Mr.**T.Anil Kumar,** for valuable suggestions and keen interest throughout the progress of my course of research.

We are grateful to **CH.Lakshmibala**, HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with my project.

At the outset, We would like to thank **Rajiv Gandhi University of Knowledge and Technologies, Srikakulam** for providing all the necessary resources for the successful completion of my course work. At last, but not least we thank our classmates and other students for their

physical and moral support.

With Sincere Regards,

Priyanka Kommani	**S180124**
Geetha Chikati	**S180402**
Samdan Shaik	**S180716**
Satya Priya Telukula	**S180130**

Abstract

The current way of checking subjective paper is adverse. Evaluating the Subjective Answers is a critical task to perform. When human being evaluates anything, the quality of evaluation may vary along with the emotions of person. During the pandemic the administration had gone through a very rough cycle in conducting online exams and evaluating the answer sheets as every single paper has to be evaluated manually even the exam was in online mode. This project automates this problem. Our proposed system uses python to solve this problem. Our Algorithm performs a task like Tokenizing words and sentences, comparing the similarity of data. It retrieves data from a spreadsheet and uses NLP Techniques and Machine learning models to evaluate the subjective answer. Our System is divided into four modules. The first one is preparing the data , second one retrieving and comparing data from a spreadsheet with certain python libraries ,third one is frontend and the fourth is giving marks and detailed report to them.

List of figures:

Contents :

Chapter 1

Introduction

1.1 Introduction

In the present system manually evaluating the subjective answers was a challenge as it needs more time and human. Unlike objective questions that have only one correct answer, subjective questions require a personal opinion or interpretation, making it challenging to grade them using automated methods. A subjective answer checker is a tool or software designed to evaluate subjective responses to open-ended questions or prompts. This technology is useful for grading subjective answers in areas such as education, recruitment, and online courses, as it can save time for human evaluators while ensuring fair and consistent grading.

1.2 Applications

- Helps in educational institutions to grade open-ended questions and essays.
- Provides a comprehensive analysis of the candidate’s performance.
- Provides a better platform to reduce manual work.
- No chance of delays and postponement of evaluation.

1.3 Problem Statement

To provide a better evaluation process that works automatically which reduces human efforts and gives accurate grading for the subjective answers .

Chapter 2

Literature Survey

2.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans using natural language. It involves the development of algorithms and computational models that enable computers to understand, interpret, and generate human-like text.

NLP encompasses a wide range of tasks, including:

1. **Text Understanding:** Extracting meaning from text, including sentiment analysis, entity recognition, and topic modeling.

2. **Language Generation:** Creating coherent and contextually relevant text, such as chatbots, language translation, and content creation.

3. **Speech Recognition:** Converting spoken language into written text, enabling voice interfaces and voice commands.

4. **Information Retrieval:** Extracting relevant information from large datasets or databases based on user queries.

5. **Machine Translation:** Translating text from one language to another automatically.

6. **Text Summarization:** Generating concise and meaningful summaries of longer texts.

NLP applications are widespread and impact various industries, including healthcare, finance, customer service, and more. Techniques used in NLP include machine learning, deep learning, and linguistic rule-based approaches to analyze and understand human language in a meaningful way.

2.2 Python Language Tool Kit

Python itself is a programming language, not a toolkit. However, Python has a rich ecosystem of libraries and tools that developers commonly use for various tasks. Here are some key components of the Python ecosystem:

1. **Python Standard Library:** Python comes with a comprehensive standard library that includes modules for file I/O, networking, regular expressions, data structures, and more. These modules are part of the standard Python distribution.
2. **Integrated Development Environments (IDEs):** Popular Python IDEs include PyCharm, Visual Studio Code, Jupyter Notebooks, and IDLE. These tools provide features such as code completion, debugging, and project management.
3. **Package Managers:** Python uses package managers like pip and conda to install, manage, and distribute third-party libraries and packages.
4. **Data Science and Machine Learning Libraries:** Python is widely used in data science and machine learning. Libraries like NumPy, pandas, scikit-learn, and TensorFlow provide tools for data manipulation, analysis, and machine learning.

5. **Web Development Frameworks:** Flask and Django are popular Python frameworks for web development. They simplify the process of building web applications and APIs.

6. **Automation and Scripting:** Python is commonly used for automation tasks and scripting. Tools like Ansible leverage Python for configuration management and automation.

7. **Testing Frameworks:** Python has robust testing frameworks such as pytest

and unittest, allowing developers to write and execute tests to ensure code quality.

8. **Documentation Tools:** Sphinx is a widely used documentation tool for Python projects, enabling developers to create comprehensive documentation for their code.

9. **Version Control:** Git is often used for version control in Python projects, and platforms like GitHub and GitLab host and manage Python code repositories.

Python's versatility and extensive community support make it a popular choice for a wide range of applications, from web development to scientific computing and artificial intelligence. If you have a specific aspect or toolkit in mind, please provide more details for a more tailored response.

2.3 spacy

spacy is an open-source natural language processing (NLP) library for Python that is designed to be fast, efficient, and production-ready. Developed by Explosion AI, spacy is widely used for various NLP tasks and is known for its simplicity, ease of use, and high-performance capabilities. Here are some key features and aspects of spacy:

1. **Tokenization:** spacy provides efficient tokenization, breaking down text into words, punctuation, and other meaningful units.

2. **Part-of-Speech Tagging:** It can assign grammatical parts of speech to words in a sentence, aiding in syntactic analysis.

3. **Named Entity Recognition (NER):** spacy excels in identifying and classifying named entities such as persons, organizations, locations, dates, and

more.

4. **Dependency Parsing:** The library can analyze the grammatical structure of a sentence, providing information about the relationships between words.

5. **Pre-trained Models:** spacy comes with pre-trained models for multiple languages, which are trained on large corpora and can be fine-tuned for specific tasks.

6. **Word Embeddings:** It supports word embeddings and similarity calculations, allowing for word vectorization and semantic similarity analysis.

7. **Entity Linking:** spacy includes features for linking named entities to knowledge bases like Wikipedia, aiding in deeper understanding.

8. **Text Classification:** It supports text classification tasks, enabling users to train models for sentiment analysis, spam detection, and more.

9. **Rule-based Matching:** spacy provides a rule-based matching engine, allowing users to define custom patterns and rules for extracting information from text.

10. **Integration with Other Libraries:** It seamlessly integrates with other popular Python libraries, including scikit-learn, pandas, and NumPy.

11. **Multi-Language Support:** spacy supports multiple languages and is constantly updated with improvements and language extensions.

Chapter 3

Openpyxl & Tkinter library

3.1 Openpyxl

Openpyxl is a Python library for reading and writing Excel (xlsx) files. It allows developers to work with Excel files, manipulate data, and create or modify spreadsheets. Some key features and aspects of openpyxl include:

1. **Excel File Compatibility:** openpyxl supports the xlsx file format used by Microsoft Excel. It enables the creation, modification, and extraction of data from Excel files.

2. **Read and Write Operations:** Developers can read data from existing Excel files and write data to new or existing files. This includes working with worksheets, rows, and cells.

3. **Cell Styling:** openpyxl supports the styling of cells, allowing developers to set fonts, colors, borders, and other formatting options within the Excel file.

4. **Charts and Images:** It provides functionality to insert charts and images into Excel sheets, enhancing the visual representation of data.

5. **Formula Support:** Developers can work with Excel formulas, allowing the creation of dynamic and calculated data within spreadsheets.

6. **Compatibility with Pandas:** openpyxl is often used in conjunction with the pandas library, facilitating seamless integration between data manipulation in Python and Excel.

7. **Iterative Parsing:** For large Excel files, openpyxl supports iterative parsing, enabling efficient handling of large datasets.

3.2 Tkinter Library

Tkinter is a standard GUI (Graphical User Interface) toolkit that comes with the Python programming language. It is based on the Tk GUI toolkit, which was developed by John Ousterhout. Tkinter provides a set of Python modules that allow programmers to create GUI applications with a consistent look and feel across platforms.

Architecture: Tkinter is based on the Tk toolkit, which is a cross-platform GUI toolkit written in the C programming language. Tk provides a set of widgets (such as buttons, labels, and text boxes) and a way to handle events (such as button clicks and keyboard input). Tkinter is a Python interface to the Tk toolkit, which means that it provides Python bindings to the underlying C library.

Widgets: Tkinter provides a wide range of widgets that developers can use to build their GUI applications. Some of the commonly used widgets include: Button: A widget that the user can click to trigger an action.

Label: A widget that displays text or an image. Entry: A widget that allows the user to enter text. Text: A widget that displays multiple lines of text. **Checkbutton:** A widget that allows the user to select or deselect an option. **Radiobutton:** A widget that allows the user to select one option from a group of options.

Menu: A widget that displays a list of options that the user can select. Layout Management: Tkinter provides several methods for laying out widgets within a window or frame. The most commonly used methods are:

Pack: A simple method that places widgets in a horizontal or vertical stack.

Grid: A method that organizes widgets in a grid of rows and columns.

Place: A method that allows precise positioning of widgets using x and y coordinates.

Event Handling: Tkinter provides a way to handle events, such as button clicks or key presses. Developers can define event handler functions that are called when an event occurs. For example, to handle a button click event, a developer can define a function that is called when the button is clicked.

Themes: Tkinter provides several built-in themes that allow developers to change the appearance of their GUI applications. Developers can also create custom themes using the Tkinter theming engine.

Compatibility: Tkinter is compatible with Python 2 and Python 3. However, there are some differences in the way that Tkinter is used between the two versions. For example, in Python 3, Tkinter uses the unicode character set by default, while in Python 2, it uses the ASCII character set.

To create a GUI application using Tkinter,

- a developer typically follows these steps:
- Import the Tkinter module.
- Create a root window.
- Add widgets to the root window.
- Set properties for the widgets.
- Define event handlers for the widgets.
- Enter the main event loop.

Chapter 4

Approach to Descriptive answer evaluation.

4.1 Introduction

The provided Python script is designed for processing and analyzing answers in an Excel spreadsheet, particularly for keyword extraction and generating an answer key. Here's a detailed description of its key components:

4.2 Libraries and Modules:

- **`openpyxl`:** Used for interacting with Excel files, allowing the script to read and manipulate spreadsheet data.
- **`utils`:** Presumed to contain utility functions or variables, though the specific content is not provided in the given code snippet.
- **`spacy`:** A natural language processing library that is used for various text processing tasks, including the analysis of textual data.
- **`stopwords`:** From the NLTK library, providing a set of common English stopwords for later use in text processing.
- **`pytextrank`:** Implements the TextRank algorithm, enhancing spacy's capabilities for keyword extraction in the text.
- **`yake`:** A library for keyword extraction using the YAKE (Yet Another Keyword Extractor) algorithm.
- **`en_core_web_sm`:** The English language model for spacy, which is loaded to perform natural language processing tasks.

4.3 Loading Spacy Model and Adding TextRank:

- The script loads the spacy English language model (`en_core_web_sm`), enabling it to understand and process English text.
- It enhances spacy's capabilities by adding the TextRank algorithm to the spacy pipeline. This is done to extract essential keywords from the text later in the script.

4.4 Answer Key Generation Function (`ans_key`):

- Clears and initializes lists from the `utils` module, indicating the script may depend on external utility functions or variables.
- Calls the `get_anslst` function to retrieve a list of answers from a specified Excel sheet.
- Calls the `print_ans` function to convert the answer list into a list of strings.
- Calls the `get_keywords` function to extract keywords from the obtained answers.

4.5 Answer List Extraction Function (`get_ansist`):

- Accepts an Excel sheet as input.
- Iterates through the rows of the sheet to extract answers grouped together.
- Returns a list of lists, where each inner list contains a group of related answers.

4.6 Keyword Extraction Function (`get_keywords`):

- Utilizes the YAKE library for keyword extraction from each group of answers.

- Appends the extracted keywords to a list for each group of answers.
- Returns a list of lists, where each inner list contains the extracted keywords for a particular group of answers.

4.7 Print Answer Function (`print_ans`):

- Converts the answer list into a list of strings, where each string represents a group of answers.
- Returns a list of strings, providing a formatted representation of the answers.

4.8 Comments and Notes:

- The script includes commented-out print statements, which could be useful for debugging or understanding the intermediate steps.
- Several functions and modules, such as `utils`, `get_anslst`, `get_keywords`, `print_ans`, are referenced but their actual implementation is not provided.
- The script seems designed to handle multiple groups of answers, potentially corresponding to different questions or sections in a survey or test.
- The functionality and purpose of the script would be clearer with the content of the missing parts, especially those from the `utils` module.

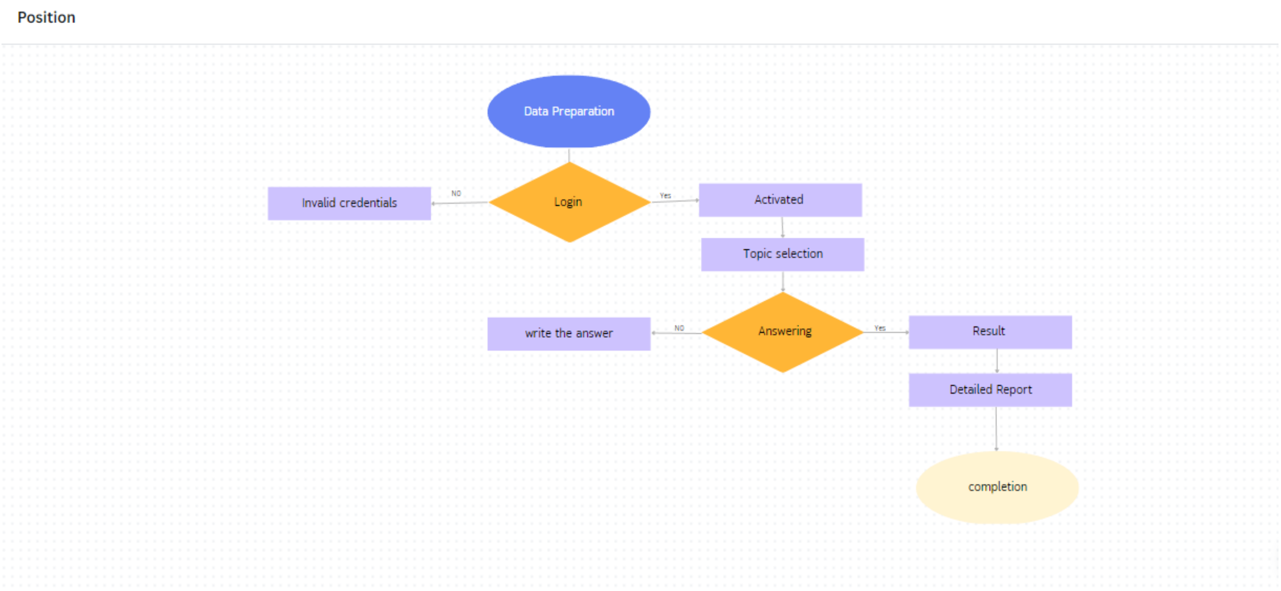
Chapter 5

Procedure Implementation and Results

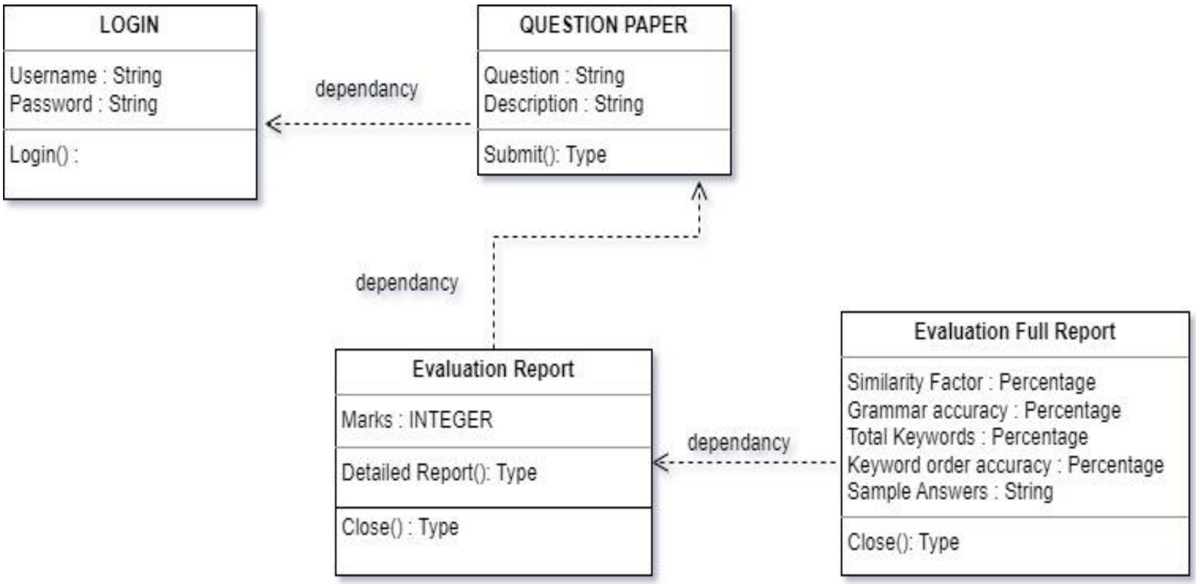
5.1 Procedure

We divided the process of implementation into parts. First among them is descriptive answer to keywords and Text implementation, Then we worked on descriptive answer evaluation which is the core of our project as it is important to evaluate the answer of the user, We further worked on question generation from the stored data, we used keyword comparison to implement this and Our final work is to calculate the average score to decide when the user should end and also to the display the evaluation report.

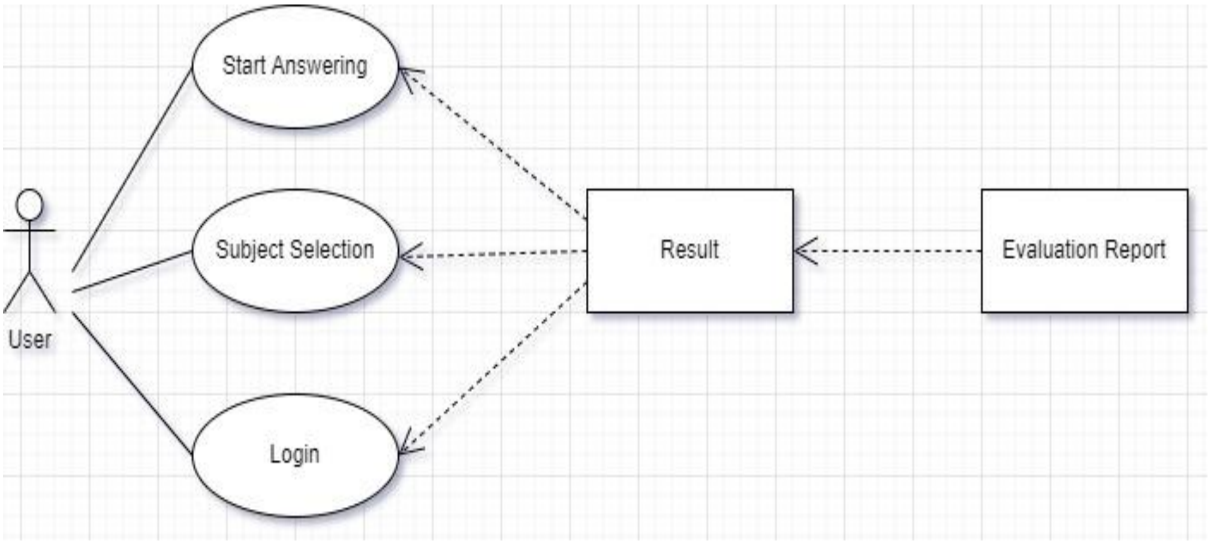
5.2 Flowchart



5.3 Class Diagram



5.4 Use Case Diagram



5.5 Implementation

5.5.1Code

```
import tkinter as tk
from tkinter import *
import openpyxl
from Test import Test
from openmyfile import openmyfile
import utils
from PIL import ImageTk, Image
from tkinter import ttk
from tkinter import Tk, Toplevel, StringVar, Label, Entry, Frame, OptionMenu, Button, messagebox
class SignUp(tk.Toplevel):
    def __init__(self, master=None):
        tk.Toplevel.__init__(self, master)
        self.title("Sign Up")
        self.resizable(False, False)
        self.tk_setPalette(background='#fff')
        self.geometry("925x500+300+200")
        # Add the image
        img_path = 'signup.png'
        self.display_image(img_path, 75, 50)
        frame = Frame(self, width=350, height=350, bg="white")
        frame.place(x=480, y=70)
        heading = tk.Label(frame, text='Sign Up', fg="#57a1f8", bg='white', font=('Comic Sans MS', 30, 'bold'))
        heading.place(x=100, y=5)
        # username
        def on_leave_username(e):
            name = self.user.get()
            if name == '':
                self.user.delete(0, tk.END)
                self.user.insert(0, 'Username')
        def on_enter_username(e):
            self.user.delete(0, tk.END)

        self.user = Entry(frame, width=25, fg='black', border=0, bg='white', font=('Microsoft YaHei UI Light', 11))
        self.user.place(x=30, y=80)
        self.user.insert(0, 'Username')
        self.user.bind('<FocusIn>', on_enter_username)
        self.user.bind('<FocusOut>', on_leave_username)
        Frame(frame, width=295, height=2, bg='black').place(x=25, y=107)
        # passcode
        def on_leave_passcode(e):
            name = self.passcode.get()
            if name == '':
                self.passcode.delete(0, tk.END)
                self.passcode.insert(0, 'Password')
        def on_enter_passcode(e):
            self.passcode.delete(0, tk.END)
        self.passcode = Entry(frame, width=25, fg='black', border=0, bg='white', font=('Microsoft YaHei UI Light', 11))
        self.passcode.place(x=30, y=130)
        self.passcode.insert(0, 'Password')
        self.passcode.bind('<FocusIn>', on_enter_passcode)
        self.passcode.bind('<FocusOut>', on_leave_passcode)
        Frame(frame, width=295, height=2, bg='black').place(x=25, y=157)
        # confirm passcode
        def on_leave_conf_passcode(e):
            name = self.conf_passcode.get()
            if name == '':
                self.conf_passcode.delete(0, tk.END)
                self.conf_passcode.insert(0, 'Confirm Password')
        def on_enter_conf_passcode(e):
```

```
            self.conf_passcode.delete(0, tk.END)
        self.conf_passcode = Entry(frame, width=25, fg='black', border=0, bg='white', font=('Microsoft YaHei UI Light', 11))
        self.conf_passcode.place(x=30, y=180)
        self.conf_passcode.insert(0, 'Confirm Password')
        self.conf_passcode.bind('<FocusIn>', on_enter_conf_passcode)
        self.conf_passcode.bind('<FocusOut>', on_leave_conf_passcode)
        Frame(frame, width=295, height=2, bg='black').place(x=25, y=207)
        Button(frame, width=39, pady=7, text='Sign Up', bg='#57a1f8', fg='white', border=0, command=self.click_submit).place(x=35, y=234)
        label = Label(frame, text="Existing account? ", fg="black", bg="white", font=('Microsoft YaHei UI Light', 10, 'bold'))
        label.place(x=75, y=280)
        login = Button(frame, width=6, text="Sign In", border=0, bg='white', cursor='hand2', fg="#57a1f8",
                     font=('Microsoft YaHei UI Light', 9, 'bold'), command=self.click_signin)
        login.place(x=245, y=280)
    def click_submit(self):
        if self.passcode.get() == self.conf_passcode.get():
            user = self.user.get()
            password = self.passcode.get()
        else:
            messagebox.showinfo("Signup error", "Password and confirm password do not match!")
            return
        op = "users.xlsx"
        workbook = openpyxl.load_workbook(op)
        sheet = workbook.active
        next_row = sheet.max_row + 1
        for i in range(2, next_row):
            if sheet.cell(row=i, column=1).value == user:
                messagebox.showinfo("Signup error", "User already exists!")
                break
        else:
            sheet.cell(row=next_row, column=1).value = user
            sheet.cell(row=next_row, column=2).value = password
            workbook.save(op)
            utils.user = user
            # #answers.xlsx
            # workbook = openpyxl.load_workbook("answers.xlsx")
            # sheet = workbook.active
            # next_row = sheet.max_row + 1
            # sheet.cell(row=next_row, column=1).value = user
            # workbook.save(op)
            self.newWindow = Test()
            self.destroy()
            root.withdraw()
    def display_image(self, img_path, x, y):
        img = Image.open(img_path)
        img = img.resize((300, 300), Image.LANCZOS)
        img = ImageTk.PhotoImage(img)
        img_label = Label(self, image=img, bg='white')
        img_label.image = img  # Keep a reference to prevent garbage collection
        img_label.place(x=x, y=y)
    def click_signin(self):
        self.destroy()  # Destroy the sign-up window
        root.deiconify()  # Show the existing sign-in window
class App(tk.Frame):
    def __init__(self):
        tk.Frame.__init__(self)
        self.pack()
        self.master.tk_setPalette(background='white')
        self.master.geometry('925x500+300+200')
        self.master.title("Sign In")
        # Create a Tkinter variable
        tkt = StringVar(root)
```

```
global img
# Add the image
img_path = 'login.png'
self.display_image(img_path, 75, 50)
frame = Frame(self.master, width=350, height=350, bg="white")
frame.place(x=480, y=70)
heading = Label(frame, text='Sign In', fg="#57a1f8", bg='white', font=('Comic Sans MS', 30, 'bold'))
heading.place(x=100, y=5)
# username
def on_leave_username(e):
    name = self.user.get()
    if name == '':
        self.user.delete(0, tk.END)
        self.user.insert(0, 'Username')
def on_enter_username(e):
    self.user.delete(0, tk.END)
self.user = Entry(frame, width=25, fg='black', border=0, bg='white', font=('Microsoft YaHei UI Light', 11))
self.user.place(x=30, y=80)
self.user.insert(0, 'Username')
self.user.bind('<FocusIn>', on_enter_username)
self.user.bind('<FocusOut>', on_leave_username)
Frame(frame, width=295, height=2, bg='black').place(x=25, y=107)
# passcode
def on_leave_passcode(e):
    name = self.passcode.get()
    if name == '':
        self.passcode.delete(0, tk.END)
        self.passcode.insert(0, 'Password')
def on_enter_passcode(e):
    self.passcode.delete(0, tk.END)
self.passcode = Entry(frame, width=25, fg='black', border=0, bg='white',
font=('Microsoft YaHei UI Light', 11))
        self.passcode.place(x=30, y=130)
        self.passcode.insert(0, 'Password')
        self.passcode.bind('<FocusIn>', on_enter_passcode)
        self.passcode.bind('<FocusOut>', on_leave_passcode)
        Frame(frame, width=295, height=2, bg='black').place(x=25, y=157)
        # Dictionary with options
        # choices = {'E-Commerce', 'NLP', 'Cryptography', 'Cyber-Security',
'Philosophy'}
        # tkt.set('Cyber-Security')  # set the default option
        # openmyfile('Cyber-Security')
        # popupMenu = OptionMenu(frame, tkt, *choices)
        # subj = Label(frame, text="Subject       : ", font=('Microsoft YaHei UI
Light', 11))
        # subj.place(x=30, y=180)
        # popupMenu.place(x=140, y=180)
        Button(frame, width=39, pady=7, text='Sign In', bg='#57a1f8', fg='white',
border=0,
               command=self.click_ok).place(x=35, y=234)
        label = Label(frame, text="Don't have an account?", fg="black",
bg="white",
                      font=('Microsoft YaHei UI Light', 10, 'bold'))
        label.place(x=75, y=280)
        sign_up = Button(frame, width=6, text="Sign Up", border=0, bg='white',
cursor='hand2', fg="#57a1f8",
                        font=('Microsoft YaHei UI Light', 9, 'bold'),
command=self.click_signup)
        sign_up.place(x=245, y=280)
        combo = ttk.Combobox(frame, textvariable = tkt, values= ['E-Commerce',
'NLP', 'Cryptography', 'Cyber-Security', 'Philosophy'])
        combo.set('Cyber-Security')
        openmyfile('Cyber-security')
        subj = Label(frame, text="Choose subject    : ", font=('Microsoft YaHei UI
Light', 11, ), bg='white')
```

```
        subj.place(x=30, y=180)
        combo.place(x=180, y=180)
        def change_dropdown(*args):
            utils.file_name = tkt.get()
            openmyfile(utils.file_name)
        # link function to change dropdown
        tkt.trace('w', change_dropdown)
    def click_ok(self):
        user = self.user.get()
        password = self.passcode.get()
        op = "users.xlsx"
        workbook = openpyxl.load_workbook(op)
        sheet = workbook.active
        next_row = sheet.max_row + 1
        found=False
        for i in range(2, next_row):
            if sheet.cell(row=i, column=1).value == user and sheet.cell(row=i, column=2).value == password:
                found=True
                utils.user = user
                self.newWindow = Test()
                root.withdraw()
                break
            elif sheet.cell(row=i, column=1).value == user and sheet.cell(row=i, column=2).value != password:
                found=True
                messagebox.showinfo("Sign In error", "Incorrect password!")
        if found==False:
            messagebox.showinfo("Sign In error", "User does not Exist! Please SignUp!")
    def display_image(self, img_path, x, y):
        img = Image.open(img_path)
        img = img.resize((300, 300), Image.LANCZOS)
        img = ImageTk.PhotoImage(img)
        img_label = Label(self.master, image=img, bg='white')
        img_label.image = img  # Keep a reference to prevent garbage collection
        img_label.place(x=x, y=y)
    def click_signup(self):
        self.newWindow = SignUp()
        root.withdraw()


if __name__=='__main__':
    root = Tk()
    app = App()
    root.mainloop()
```

5.5.2 Pre-Processed Questions and Answers:

Q1: What is SHA ?

SHA, which stands for Secure Hash Algorithm, is a cryptographic hash function widely used in computer security and digital signatures. It takes an input (message) and produces a fixed-size hash value, typically represented as a sequence of alphanumeric characters. The SHA algorithm ensures that even a slight change in the input message will result in a significantly different hash value, providing data integrity and security.

SHA, or Secure Hash Algorithm, is a family of cryptographic hash functions developed by the National Security Agency (NSA). It is used to generate a unique fixed-size hash value for a given input, regardless of the input's size. This makes SHA useful for verifying the integrity of data, detecting duplicates, and securely storing passwords. Common variants of SHA include SHA-1, SHA-256, and SHA-512, each with different hash lengths and levels of security.

SHA, short for Secure Hash Algorithm, is a widely used cryptographic hash function designed to ensure the integrity of data. It takes an input message and produces a fixed-length hash value that is unique to the input. The SHA algorithm is characterized by its one-way nature, meaning it is computationally infeasible to derive the original message from the hash value. This property makes SHA suitable for verifying data integrity, password hashing, digital signatures, and various other applications in information security.

Q2: How does SHA contribute to data integrity and security in cryptographic systems?

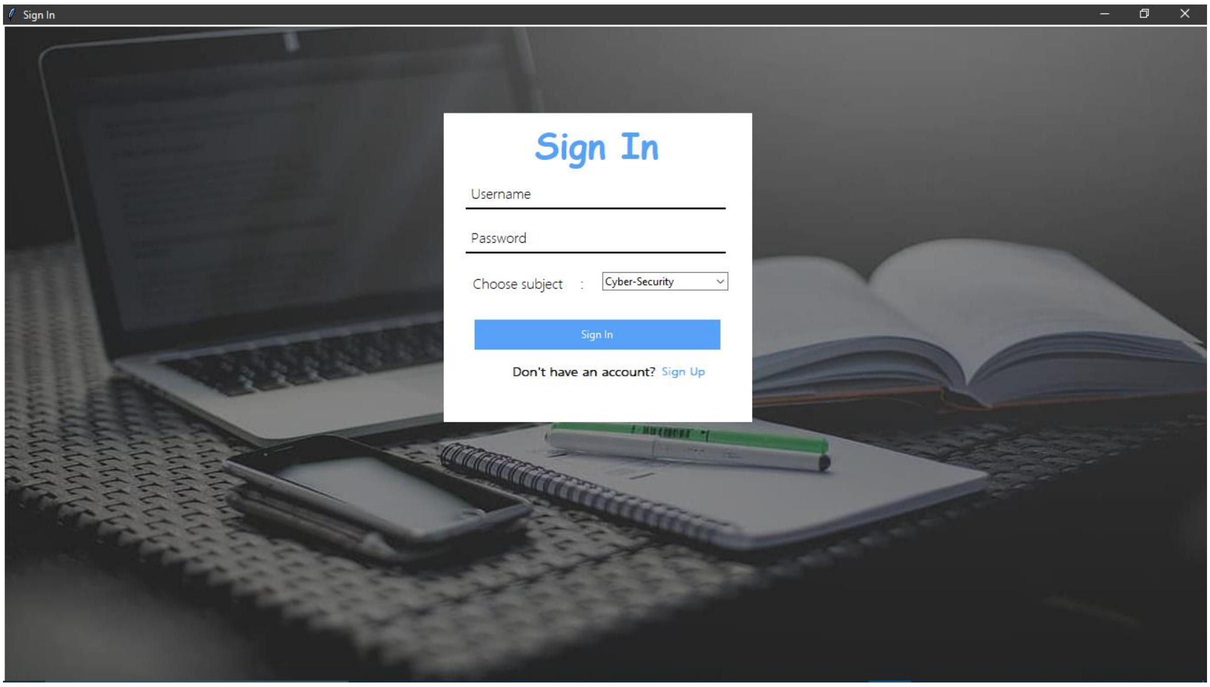
SHA plays a crucial role in ensuring data integrity and security in cryptographic systems. By generating a unique hash value for a given input, SHA allows users to verify that data has not been tampered with during transmission or storage. Even a small change in the input message will result in a drastically different hash value, providing a strong indication of data alteration.

The application of SHA in cryptographic systems helps in detecting and preventing unauthorized modifications to data. By comparing the hash value of received data with the original hash value, users can ensure the integrity of the information. This prevents attackers from tampering with the data unnoticed, as any changes will lead to a mismatch in the hash values.

In cryptographic systems, SHA contributes to data integrity and security by acting as a digital fingerprint for the input data. The hash value generated by SHA is unique to the input message, making it extremely unlikely for two different messages to produce the same hash. This property ensures that even a small change in the input will produce a distinct hash, providing a reliable means of detecting data tampering.

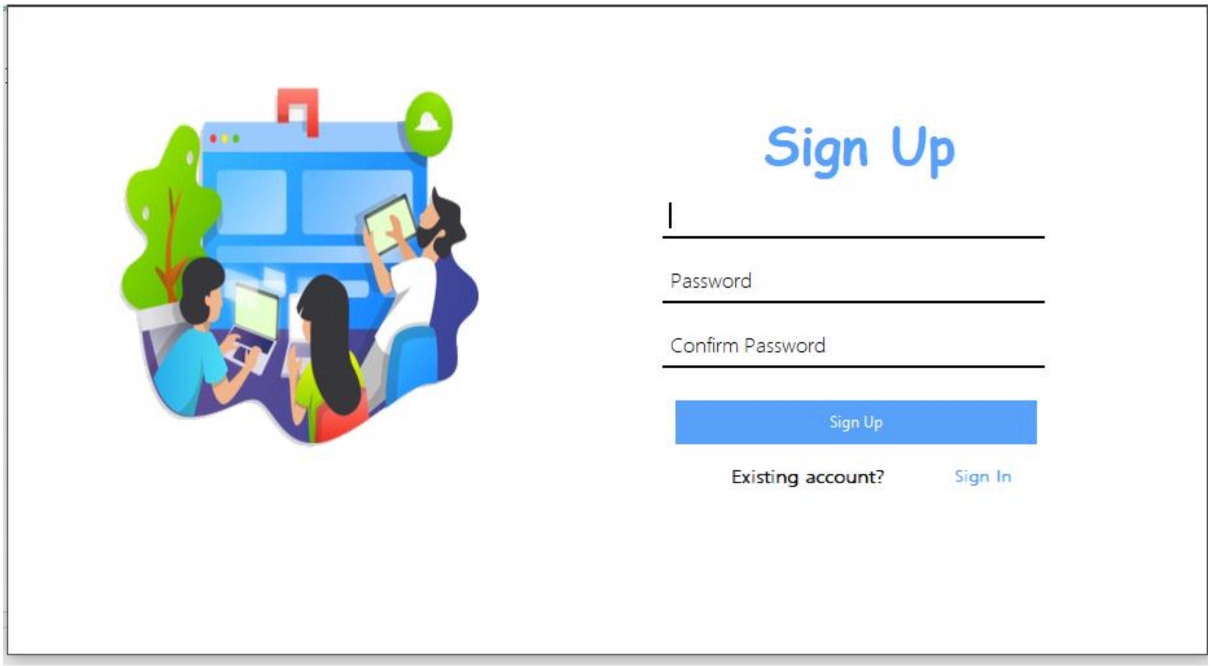
5.3 Results

Sign in Page:

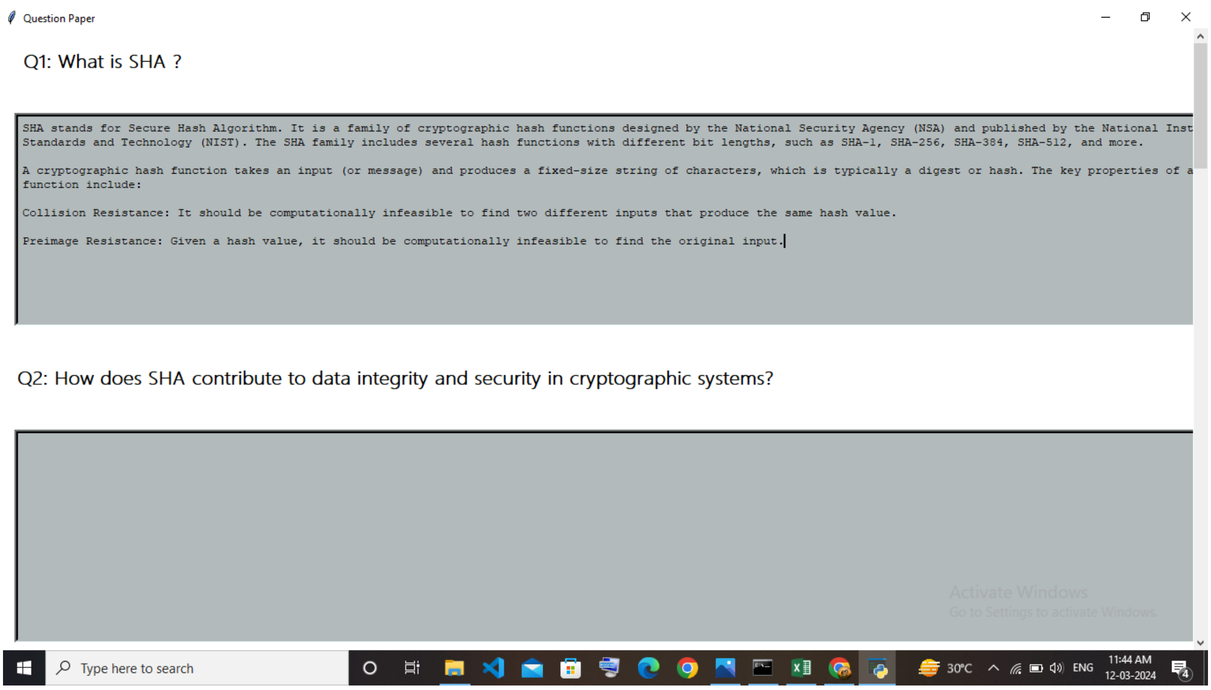


Choose subject : Cyber-Security

Sign Up page:



Question paper

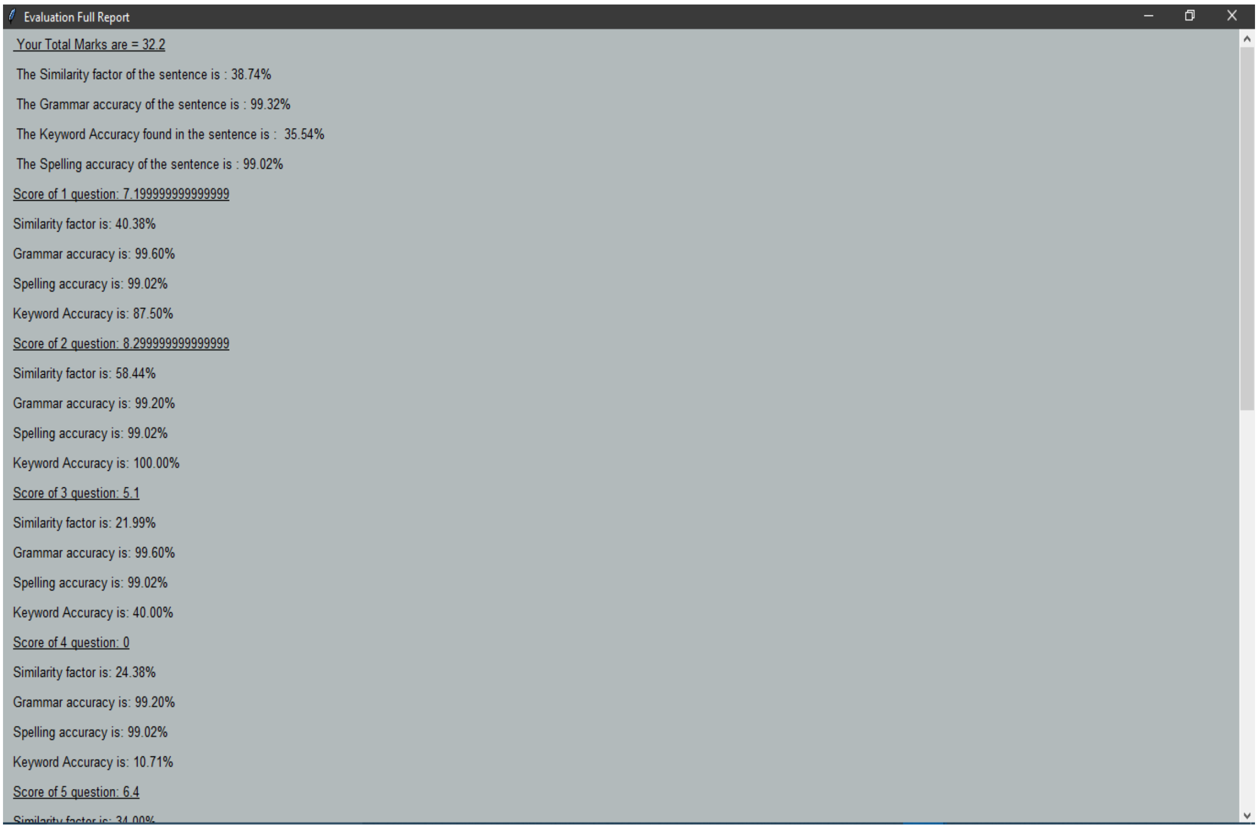


Q1: What is SHA ?

```
SHA stands for Secure Hash Algorithm. It is a family of cryptographic hash functions designed by the National Security Agency (NSA) and published by the National Inst
Standards and Technology (NIST). The SHA family includes several hash functions with different bit lengths, such as SHA-1, SHA-256, SHA-384, SHA-512, and more.

A cryptographic hash function takes an input (or message) and produces a fixed-size string of characters, which is typically a digest or hash. The key properties of a
function include:

Collision Resistance: It should be computationally infeasible to find two different inputs that produce the same hash value.

Preimage Resistance: Given a hash value, it should be computationally infeasible to find the original input.
```

Q2: How does SHA contribute to data integrity and security in cryptographic systems?

Report



```
 Your Total Marks are = 32.2
 The Similarity factor of the sentence is : 38.74%
 The Grammar accuracy of the sentence is : 99.32%
 The Keyword Accuracy found in the sentence is :  35.54%
 The Spelling accuracy of the sentence is : 99.02%
Score of 1 question: 7.199999999999999
Similarity factor is: 40.38%
Grammar accuracy is: 99.60%
Spelling accuracy is: 99.02%
Keyword Accuracy is: 87.50%
Score of 2 question: 8.299999999999999
Similarity factor is: 58.44%
Grammar accuracy is: 99.20%
Spelling accuracy is: 99.02%
Keyword Accuracy is: 100.00%
Score of 3 question: 5.1
Similarity factor is: 21.99%
Grammar accuracy is: 99.60%
Spelling accuracy is: 99.02%
Keyword Accuracy is: 40.00%
Score of 4 question: 0
Similarity factor is: 24.38%
Grammar accuracy is: 99.20%
Spelling accuracy is: 99.02%
Keyword Accuracy is: 10.71%
Score of 5 question: 6.4
Similarity factor is: 34.00%
```

Chapter 6

Limitations and Extensions

6.1 Limitations

The main limitations of the proposed system are:

- Specific to English language only.
- Requires specific system requirements.
- Does not evaluate answers with diagrams and mathematical expressions. · Polarity of the sentences is not considered.

6.2 Extensions

- Multimodal evaluation: Many written responses include additional forms of expression beyond text, such as images or videos.
- Incorporating natural language generation: The use of natural language generation (NLG) techniques could enable subjective answer checkers to provide more detailed and informative feedback to learners, including suggestions for improvement or links to additional resources.
- Supporting more languages: While most current subjective answer checkers are designed for English-language responses, there is a growing need for tools that can support responses in other languages. Future extensions could include the development of scoring systems for a wider range of languages.

- Incorporating human feedback: Automated scoring systems could be further improved by incorporating feedback from human experts, either through the use of crowdsourcing or by using machine learning techniques to learn from expert feedback.

Chapter7

Testing

Unit testing, Integration testing, System testing, and Acceptance testing are all different types of software testing, each with its own objectives and characteristics. Here's an overview of each type:

Unit Testing:

Unit testing is a testing approach where individual units or components of software are tested in isolation from the rest of the system. The objective of unit testing is to verify that each unit of code functions as intended and meets its requirements. It is typically performed by developers and is focused on identifying and fixing bugs at an early stage of the software development cycle.

Login credentials testing, Questions retrieve testing, written answer testing, Result testing, detail report testing are the individual testing done in our project.Login credentials testing is whether the user giving correct credentials or not,written testing is whether the user wrote answer or not , Result testing is whether the given result gives correct marks or not.

Integration Testing:

Integration testing is a testing approach that verifies the interactions between different software components or subsystems. The objective of integration testing is to ensure that the different parts of the system work together correctly and as expected. Integration testing can be performed at different levels, such as component integration testing, where individual components are tested together, and system integration testing, where the entire system is tested as a whole.

Integration testing is interaction between Login credentials testing, Questions retrieve testing, written answer testing, Result testing, detail report testing , in the project every module system work together correctly.

System Testing:

System testing is a testing approach that evaluates the overall system behavior in terms of meeting its specified requirements. It is typically performed after integration testing and before acceptance testing. The objective of system testing is to ensure that the system meets the functional and non-functional requirements and works as expected in its intended environment.

project successfully passed system testing where overall system behaviour worked in expected way, while running overall code there is no problem with any of the outputs.

Acceptance Testing:

Acceptance testing is a testing approach that validates the system against the requirements and expectations of the end-users or customers. The objective of acceptance testing is to ensure that the system meets the user's needs and is ready for deployment. Acceptance testing can be performed by end-users, customers, or a dedicated acceptance testing team.

Project passed all the requirements and expectations of our guide, project is ready to Deploy in certain sectors like education sector, ecommerce sector, industrial sector etc.

Chapter 8
Conclusion

Automated Paper Evaluation are automated software tools designed to evaluate the quality of written responses that require a subjective evaluation, such as essays and open-ended questions. Subjective answer checkers have the potential to save time and effort compared to manual evaluation by human experts, and they can provide more consistent and objective evaluations. However, these tools also have some limitations, such as their inability to evaluate some aspects of the response, such as creativity or originality, and their potential to be influenced by differences in writing styles or cultural backgrounds. Despite these limitations, Automated Paper Evaluation will show better results, with automated scoring systems achieving a high level of accuracy.

Bibliography

- Evaluating Students’ Descriptive Answers Using Natural Language Processing And Artificial Neural Networks V. Lakshmi, V. Ramesh
- Dikli, S. (2006). An overview of automated scoring of essays. The Journal of Technology, Learning, and Assessment.
- Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater® V.2. Journal of Technology, Learning, and Assessment, Vajjala, S., & Meurers, D. (2012). An empirical study on the effectiveness of using supervised machine learning for holistic scoring of non-native spontaneous speech. CALICO Journal, 29(3), 542-562