AUTOMATED ANSWER EVALUATION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

ASHOK AT	**510420104015**
AJAYDEV MS	**510420104009**
CHANDRU V	**510420104020**
AASHIP J	**510420104001**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ARUNAI ENGINEERING COLLEGE
TIRUVANNAMALAI- 606603

ANNA UNIVERSITY:: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report **"A NEW METHOD FOR AUTOMATED ANSWER EVALUATION USING MACHINE LEARNING"** is the bonafide work of **"ASHOK AT(510420104015), AJAYDEV MS (510420104009), CHANDRU V (510420104020), AASHIP J(510420104001)"** who carried out the project work under my supervision.

SIGNATURE OF HOD
Mrs. V. UMADEVI, M.E.,
HEAD OF THE DEPARTMENT,
Computer Science and Engineering,
Arunai Engineering College,
Tiruvannamalai-606 603.

SIGNATURE OF SUPERVISOR
Ms. E. SARANYA, M.E.,
ASSISTANT PROFESSOR,
Computer Science and Engineering,
Arunai Engineering College,
Tiruvannamalai-606 603.

INTERNAL EXAMINER

EXTERNAL EXAMINER

CERTIFICATE OF EVALUATION

COLLEGE NAME : 5104-ARUNAI ENGINEERING COLLEGE

BRANCH : B.E. COMPUTER SCIENCE AND ENGINEERING

SEMESTER : 8TH SEMESTER (2020-2024)

SUBJECT CODE & NAME : CS8811 & PROJECT WORK

DATE OF EXAMINATION :

NAME OF THE STUDENTS	REGISTER NUMBER	TITLE OF THE PROJECT	NAME OF THE GUIDE WITH DESIGNATION
ASHOK AT AJAYDEV MS CHANDRU V AASHIP J	510420104015 510420104009 510420104020 510420104001	AUTOMATED ANSWER EVALUATION USING MACHINE LEARNING	Ms. E.SARANYA, M.E., Assistant Professor, Department of CSE.

The report of the **PROJECT WORK** submitted for the fulfillment of Bachelor of Engineering degree in **COMPUTER SCIENCE ENGINEERING** of Anna University was evaluated and confirmed to be reports of the work done by the above students.

SIGNATURE OF HOD
Mrs. V. UMADEVI, M.E.,
HEAD OF THE DEPARTMENT,
Computer Science and Engineering,
Arunai Engineering College,
Tiruvannamalai-606 603.

SIGNATURE OF SUPERVISOR
Ms. E. SARANYA, M.E.,
ASSISTANT PROFESSOR,
Computer Science and Engineering,
Arunai Engineering College,
Tiruvannamalai-606 603.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

It is our duty to thank the GOD Almighty and my beloved parents and teachers for their persistent blessing and constant encouragement to reach this levelof what I am today.

A project of this nature needs co-operation and support from many for successful completion. In this regard I am fortunate to express my heartfelt thanks toour beloved Founder Chairman **Mr. E.V.VELU**, Chair Person **Mrs. SANKARI VELU** and **Er. E.V.KUMARAN M.E.**, Vice Chairman, for providing necessary facilities with high-class environment throughout the course.

We take the privilege of expressing my sincere thanks to our beloved principal, **Dr.R.RAVICHANRAN Ph.D,** for granting permission to undertake the project and I also express my heartfelt thanks to Registrar **Dr.R.SATHIYASEELAN Ph.D**., for their advice in accomplishing this project work.

We are most grateful to **Mrs V.UMADEVI M.E,** Head of the Department, Department of computer Science and Engineering, who has given me both moral and technical support adding experience to the job we have undertaken.

We thank to our project guide to **Ms. E. SARANYA M.E**, Assistant Professor, Department of Computer Science and Engineering, for her astonishing guidance and encouragement for the successful completion of this project. They held me to the highest standards of quality and accuracy.

Last but not least I would like to thank my family members, friends, teaching and non-teaching staffs, who has assisted us directly or indirectly throughout this project.

ABSTRACT

AUTOMATED ANSWER EVALUATION USING MACHINE LEARNING

Nowadays, as we are moving towards automation there is a need for an automatic descriptive answer evaluation system. Manual evaluation is a time and energy-consuming task. Presently, we have automated systems for objective type, single sentence answers and descriptive answers with less accuracy level. Automatic evaluation of answer scripts has been found very useful from our experiments, and often the assigned marks is the same as manually scored marks. In this paper, our goal is to develop an automated answer evaluation system using machine learning. The system will evaluate the answer based upon the number of words and number of letters from extracted text from the pre-processed data. After that, we have to implement the Natural Language Processing (NLP) for cleaning the extracted text. The text summarization is a process of creating a short, accurate summary of the longer text. Then, we have to implement a deep learning algorithm such as Artificial Neural Network (ANN). The experimental results will be shown like accuracy and marks based on a number of words and number of letters from 0 to 10. In addition to word and letter count, our system incorporates advanced NLP techniques for deeper understanding of the text, including semantic analysis and contextual understanding. Our experiments demonstrate the effectiveness of the proposed system in accurately assessing descriptive answers, marking a significant advancement in automated evaluation methods.

திட்டச்சுருக்கம்

எம்எல் ஐப் பயன்படுத்தி தானியங்கி பதில் மதிப்பீடு

இப்போதெல்லாம், நாம் ஆட்டோமேஷனை நோக்கி நகர்ந்து வருவதால், தானியங்கி விளக்கமான பதில் மதிப்பீட்டு முறையின் தேவை உள்ளது. கைமுறை மதிப்பீடு என்பது நேரத்தையும் ஆற்றலையும் செலவழிக்கும் பணியாகும். தற்போது, எங்களிடம் புறநிலை வகை, ஒற்றை வாக்கிய பதில்கள் மற்றும் குறைவான துல்லிய நிலை கொண்ட விளக்கமான பதில்களுக்கான தானியங்கு அமைப்புகள் உள்ளன. பதில் ஸ்கிரிப்ட்களின் தானியங்கு மதிப்பீடு எங்கள் சோதனைகளிலிருந்து மிகவும் பயனுள்ளதாக இருந்தது, மேலும் பெரும்பாலும் ஒதுக்கப்பட்ட மதிப்பெண்கள் கைமுறையாக மதிப்பெண்களைப் போலவே இருக்கும். இந்த தாளில், இயந்திர கற்றலைப் பயன்படுத்தி தானியங்கு பதில் மதிப்பீட்டு முறையை உருவாக்குவதே எங்கள் குறிக்கோள். முன் செயலாக்கப்பட்ட தரவுகளிலிருந்து பிரித்தெடுக்கப்பட்ட உரையிலிருந்து வார்த்தைகளின் எண்ணிக்கை மற்றும் எழுத்துக்களின் எண்ணிக்கை ஆகியவற்றின் அடிப்படையில் கணினி பதிலை மதிப்பிடும். அதன் பிறகு, பிரித்தெடுக்கப்பட்ட உரையை சுத்தம் செய்ய இயற்கை மொழி செயலாக்கத்தை (NLP) செயல்படுத்த வேண்டும். உரை சுருக்கம் என்பது நீண்ட உரையின் குறுகிய, துல்லியமான சுருக்கத்தை உருவாக்கும் செயல்முறையாகும். பின்னர், செயற்கை நரம்பியல் நெட்வொர்க் (ANN) போன்ற ஆழமான கற்றல் வழிமுறையை செயல்படுத்த வேண்டும். சோதனை முடிவுகள் துல்லியம் மற்றும் 0 முதல் 10 வரையிலான எழுத்துக்களின் எண்ணிக்கை மற்றும் 0 முதல் 10 வரையிலான எழுத்துக்களின் அடிப்படையில் துல்லியம் மற்றும் மதிப்பெண்கள் என காண்பிக்கப்படும்.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	BONAFIDE CERTIFICATE	II
	CERTIFICATE OF EVALUATION	III
	ACKNOWLEDGEMENT	IV
	ABSTRACT ENGLISH	V
	ABSTRACT TAMIL	VI
	TABLE OF CONTENT	VII
	LIST OF FIGURES	X
	LIST OF ABREVATIONS	XI
	INTRODUCTION	
1	1.1 GENERAL	1
	1.2 LITERATURE SURVEY	2
	SYSTEM ANALYSIS	
	2.1 EXISTING SYSTEM	4
	2.1.1 DISADVANTAGES	
	2.2 PROPOSED SYSTEM	5
	2.2.1 ADVANTAGES	
	2.3 ALGORITHM	6
2	2.3.1 DECISION TREE ALGORITHM	
	2.4 SYSTEM REQUIREMENTS	
	2.4.1 HARDWARE REQUIREMENTS	
	2.4.2 SOFTWARE REQUIREMENTS	
	2.5 LIBRARY	7
	2.5.1 TKINTER	
	2.5.1.1 WHERE IS PYTHON TKINTER USED?	
	2.5.2 PYPDF2	8
	2.5.2.1 MYSQL CONNECTOR	

3	**MODULE DESCRIPTION** 3.1 ANSWER SHEET PDF MODULE 3.2 STUDENTS DETAILS ANALYSIS MODULE 3.3 STUDENTS SHEET ANALYSIS MODULE 3.4 MARK EVALUATE MODULE	 9 10
4	**SYSTEM DESIGN** 4.1 ARCHITECTURE DESIGN 4.2 AUTOMATED ANSWER EVALUATION 4.3 GUI INPUT DESIGN 4.3.1 SYSTEM DESIGN	 11 12
5	**SYSTEM DEVELOPMENT** 5.1 SOFTWARE DESCRIPTION 5.2 HISTORY OF PYTHON 5.3 PYTHON INVENTION 5.4 WORKING OF PYTHON LIBRARY 5.5 PYTHON STANDARD LIBRARY 5.6 USE OF LIBRARIES IN PYTHON 5.7 DESIGN PHILOSOPHY IN FUTURE 5.8 SYNTAX AND SEMANTICS 5.9 DEVELOPMEMT 5.9.1 API DOCUMENTATIONS GENERATORS 5.9.2 NAMING 5.10 OPEN CV 5.11 OPEN CV LIBRARY MODULES 5.11.1 CORE FUNCTIONALITY 5.11.2 IMAGE PROCESSING 5.11.3 VIDEO 5.11.4 VIDEO I/O 5.11.5 CALIB 3D 5.11.6 FEATURES 2D	 14 16 18 20 21 22 24 25

	5.11.7 OBJDETECT 5.11.8 HIGHGUI 5.12 PILLOW 5.13 TENSORFLOW 5.13.1 EASY MODEL BUILDING 5.13.2 ROBUST ML PRODUCTIO ANYWHERE 5.13.3 POWERFULL EXPERIMENTATION FOR RESEARCH 5.14 TORCH	26 27
6	**SYSTEM TESTING** 6.1 TYPES OF TEST 6.1.1 UNIT TESTING 6.1.2 INTEGRATION TESTING 6.1.3 VALIDATION TESTING	28 29
7	**SOURCE CODE AND IMPLEMENTATION**	30
8	**RESULT AND OUTPUT** 8.1 LOGIN PAGE 8.2 COLLEGE REGISTER 8.3 LOGIN SUCCESSFUL 8.4 UPLOAD STUDENT DATA 8.5 EVALUVATED SUCCESSFUL 8.6 STUDENT DATABASE	40 41 42 43
9	**CONCLUSION AND FUTURE WORK** 9.1 CONCLUSION 9.2 FUTURE WORK	44
10	**REFERENCES**	45

LIST OF FIGURES

FIGURE NO	FIGURE	PAGE NO.
4.1	ARCHITECTURE DESIGN	11
5.1	PYTHON WORKFLOW	15
5.2	FLOWCHART	23
8.1	LOGIN PAGE	40
8.2	COLLEGE REGISTER	41
8.3	LOGIN SUCCESSFUL	41
8.4	UPLOAD STUDENT DATA	42
8.5	EVALUVATED SUCCESSFUL	42
8.6	STUDENT DATABASE	43

LIST OF ABBREVATIONS

NLP - Natural Language Processing

ANN - Artificial Neural Network

TF-IDF - Term Frequency-Inverse Network Frequency

SVM - Support Vector Machines

WMD - Word Moves Distance

MNB - Multinomials Naives Bayes

OCR - Optical Character Recognition

GUI - Graphical User Interface

CWI - Centrum Wiskunde & Informatica

BDFL - Benevolent Dictator For Life

DLL - Dynamic Load Libraries

SDL - Standard DirectMedia Layer

PEP - Python Enhancement Proposal

API - Application Program Interface

OPENCV - Open source Computer Vision Library

MCQ - Multiple Choice Question

CHAPTER 1
INTRODUCTION

1.1 GENERAL

Automated answer evaluation using machine learning is a rapidly evolving field that leverages computational techniques to assess the quality and correctness of responses provided by individuals in various contexts, such as education, customer support, and technical troubleshooting. Traditional methods of evaluation often rely on human intervention, which can be time-consuming, subjective, and prone to errors. By employing machine learning algorithms, automated answer evaluation systems can analyze text-based responses and provide objective assessments based on predefined criteria or models trained on annotated datasets. These systems can range from simple rule-based models to more sophisticated deep learning architectures, depending on the complexity of the evaluation task and the available data. Feature Extraction: Transforming text inputs into numerical representations that machine learning algorithms can process. This may involve techniques such as word embeddings, TF-IDF (Term Frequency-Inverse Document Frequency), or syntactic parsing.

Model Training are utilizing annotated data to train machine learning models, which learn to recognize patterns and relationships between input answers and their corresponding evaluations. Commonly used models include Support Vector Machines (SVM), Random Forests, and neural networks.

Evaluation Metrics: Defining metrics to assess the performance of the automated evaluation system, such as accuracy, precision, recall, and F1 score. These metrics help measure how well the system aligns with human evaluations or ground truth.

1.2 LITERATURE SURVEY

TITLE: Answer Evaluation Using Machine Learning

AUTHOR: Sharad Bharadia

In this modern age, where the world moves towards automation so, there is a need for automation in answer evaluation system. Currently, the online answer evaluation is available for mcq based question, hence evaluation of the theory answer is hectic for the checker. Teacher manually checks the answer and allot the marks. The current system takes more manpower and time to evaluate the answer. In this journal an application based on the evaluation of answers using machine learning. The objective of the journal is to specially reduce the manpower and time consumption. Since in manual answer evaluation, the manpower and the time consumption is much more. Also, in the manual system, it may be possible that the marks given to two same answers are different. This application system provides an automatic evaluation of answer based on the keyword provided to the application in form of the input by the moderator which will provide equal distribution of marks and will reduce time and manpower.

TITLE: Subjective Answers Evaluation Using Machine Learning and Natural Language Processing

AUTHOR: Muhammad Farrukh Bashir, Hamza Arshad

Subjective paper evaluation is a tricky and tiresome task to do by manual labor. Insufficient understanding and acceptance of data are crucial challenges while analyzing subjective papers using Artificial Intelligence (AI). Several attempts have been made to score students' answers using computer science. However, most of the work uses traditional counts or specific words to achieve this task. Furthermore, there is a lack of curated data sets as well. This paper proposes a novel approach that utilizes various machine learning, natural language processing techniques, and tools such as Wordnet, Word2vec, word mover's distance (WMD), cosine similarity, multinomial naive bayes

(MNB), and term frequency-inverse document frequency (TF-IDF) to evaluate descriptive answers automatically. Solution statements and keywords are used to evaluate answers, and a machine learning model is trained to predict the grades of answers. Results show that WMD performs better than cosine similarity overall. With enough training, the machine learning model could be used as a standalone as well. Experimentation produces an accuracy of 88% without the MNB model. The error rate is further reduced by 1.3% using MNB.

TITLE: Automatic Answer Evaluation Using Deep Learning Algorithms

AUTHOR: Senthilkumar K, Aroabinesh J

As we go towards automation, a system for automatically evaluating descriptive responses is now required. Manual evaluation is time- and labor-intensive. Currently, our automated methods for descriptive, one-sentence, and objective responses are less accurate. The automatic scoring of answer scripts has shown to be useful in our experiments, and frequently the scores are assigned match the marks that are personally assessed. The development of an automated answer evaluation system based on machine learning is the aim of this study. The system will count the words and letters in the text that were retrieved from the pre-processed data in order to evaluate the response. The next step is to implement Natural Language Processing (NLP) to sanitize the retrieved text. Automated answer assessment is a crucial component. So, as we go towards automation, we need a framework for automatically evaluating descriptive answers. Manual evaluation requires a lot of time and effort. For objective-type, one-sentence, and descriptive answers, we currently have automated systems with lower accuracy. In our trials, the automatic scoring of answer scripts has shown to be beneficial, and frequently, the scores assigned coincide with the marks that are manually assessed. In this study, we aim to develop an automated system based on machine learning.

CHAPTER 2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In the existing grading system, teachers manually evaluate student answer booklets, which requires significant time and effort as they meticulously review each answer, provide feedback, and record grades manually. This manual process leaves room for unintentionally passing poorly written answers and failing well-performed ones.

The process often takes several months to publish results and complete the revaluation process. While there is automation for evaluating multiple-choice questions or image-based assessments, there is currently no system for evaluating answer booklets in PDF format.

2.1.1 DISADVANTAGES

Automated grading systems face several challenges, including limited understanding of nuanced responses, difficulty in handling ambiguous or creative answers, and a dependency on labeled training data, which can lead to potential bias. Moreover, these systems often struggle to assess higher-order thinking skills and face challenges in handling diverse languages or dialects.

Additionally, there is a lack of transparency in decision-making processes and difficulty in providing actionable feedback for improvement. Furthermore, automated grading systems are vulnerable to adversarial attacks or manipulation, and they may encounter scalability issues with large datasets or complex evaluation criteria. Finally, ethical concerns regarding fairness, privacy, and data security also arise in the context of automated grading systems.

2.2 PROPOSED SYSTEM

Our proposed system introduces automation for evaluating answer booklets in PDF format by leveraging Tesseract OCR, Natural Language Processing (NLP), and Machine Learning techniques. This approach allows for the automatic analysis of student responses against an answer key dataset, enabling the automatic verification of answer booklets and the assignment of grades without manual intervention.

It automates the entry of grades into students' result pages, thus avoiding administrative burden. Furthermore, students can easily download their evaluated answer booklets, verify their grades, and request reevaluation, simplifying the current revaluation process.

2.2.1 ADVANTAGES

The implementation of automated grading offers numerous benefits, including enhanced grading accuracy and consistency, which ensure fair evaluation across the board. It also reduces administrative workload and logistical costs, allowing educators to focus on more value-added tasks. With efficient access to evaluated answer booklets, students can better understand their performance and make informed decisions about their learning. The simplified reevaluation process further empowers students, promoting a culture of continuous improvement.

Overall, automated grading improves the efficiency of educational assessments, streamlining processes and saving valuable time and resources. Moreover, it increases transparency in the grading process, fostering trust and accountability among all stakeholders involved in the educational system.

2.3 ALGORITHM

2.3.1 DECISION TREE ALGORITHM

Decision trees are a type of machine-learning algorithm that can be used for both classification and regression tasks. They work by learning simple decision rules inferred from the data features. These rules can then be used to predict the value of the target variable for new data samples.

Decision trees are represented as tree structures, where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents a prediction. The algorithm works by recursively splitting the data into smaller and smaller subsets based on the feature values. At each node, the algorithm chooses the feature that best splits the data into groups with different target values.

2.4 SYSTEM REQUIREMENTS

2.4.1 HARDWARE REQUIREMENTS

- **System** : Pentium i3 Processor
- **Hard Disk** : 500 GB
- **Monitor** : 15'' LED
- **Input Devices** : Keyboard, Mouse
- **Ram** : 4 GB

2.4.2 SOFTWARE REQUIREMENTS

- **Operating system** : Windows 10.
- **Coding Language** : Python 3.8+
- **IDE** : Pycharm

2.5 LIBRARY

2.5.1 TKINTER

Tkinter is a standard Python GUI (Graphical User Interface) library that provides a set of tools and widgets to create desktop applications with graphical interfaces. Tkinter is included with most Python installations, making it easily accessible for developers who want to build GUI applications without requiring additional installations or libraries.

Full Form of Tkinter

The name “Tkinter” comes from “Tk interface“, referring to the Tk GUI toolkit that Tkinter is based on. Tkinter provides a way to create windows, buttons, labels, text boxes, and other GUI components to build interactive applications.

Significance of Tkinter

Tkinter is the inbuilt python module that is used to create GUI applications. It is one of the most commonly used modules for creating GUI applications in Python as it iis simple and easy to work with. You don’t need to worry about the installation of the Tkinter module separately as it comes with Python already. It gives an object-oriented interface to the Tk GUI toolkit. Among all, Tkinter is most widely used

2.5.1.1 Where is Python Tkinter used?

Here are some common use cases for Tkinter:

- **Creating windows and dialog boxes:** Tkinter can be used to create windows and dialog boxes that allow users to interact with your program. These can be used to display information, gather input, or present options to the user.
- **Building a GUI for a desktop application**: Tkinter can be used to create the interface for a desktop application, including buttons, menus, and other interactive elements.

- **Adding a GUI to a command-line program**: Tkinter can be used to add a GUI to a command-line program, making it easier for users to interact with the program and input arguments.
- **Creating custom widgets:** Tkinter includes a variety of built-in widgets, such as buttons, labels, and text boxes, but it also allows you to create your own custom widgets.
- **Prototyping a GUI:** Tkinter can be used to quickly prototype a GUI, allowing you to test and iterate on different design ideas before committing to a final implementation.

2.5.2 PYPDF2

PyPDF2 is a Python library for working with PDF files. It allows users to manipulate PDF documents by performing various operations such as reading, writing, merging, splitting, encrypting, and decrypting PDF files. The PyPDF2 library provides functionalities to extract text and metadata from PDF documents, manipulate individual pages, and combine multiple PDF files into a single document. With PyPDF2, users can create custom scripts or applications to automate tasks involving PDF files, such as extracting data from PDF reports, generating PDF documents dynamically, or watermarking PDF pages.

2.5.2.1 MYSQL CONNECTOR

The MySQL Connector library, often referred to as MySQL Connector/Python, is a Python library that provides an interface for interacting with MySQL databases from Python applications. It allows developers to connect to MySQL database servers, execute SQL queries, and manage database transactions directly from their Python code.MySQL Connector/Python is an implementation of the Python Database API Specification v2.0 (PEP 249), which defines a standard interface for database connectivity in Python. This means that developers can use familiar programming patterns and conventions to interact with MySQL databases, regardless of the specific Python database library they are using.

CHAPTER-3

MODULE DESCRIPTION

3.1 ANSWER SHEET PDF MODULE

Involves gathering a diverse collection of email data, comprising both spam and legitimate emails, from various sources. This data forms the basis for training and testing the spam detection model. Encompasses cleaning and preparing the collected email data for analysis. Tasks may include removing duplicate emails, handling missing values, tokenizing text, and converting categorical features into numerical representations.

3.2 STUDENT DETAILS ANALYSIS MODULE

A student detail analysis module for automated answer evaluation using machine learning involves gathering student information, extracting relevant features, preprocessing the data, integrating it with the answer evaluation system, applying machine learning algorithms to predict student performance, providing personalized feedback, tracking progress over time, visualizing data, and integrating with educational platforms. This module aims to help educators understand students' strengths, weaknesses, and areas for improvement, enabling personalized instruction and support.

3.3 ANSWER SHEET ANALYSIS MODULE

An answer sheet analysis module for automated answer evaluation using machine learning involves several essential steps. Initially, it acquires digital copies of answer sheets, either scanned or in PDF format. Then, it extracts text or handwriting from these sheets using optical character recognition (OCR) techniques. The extracted text undergoes preprocessing to clean and normalize it for analysis. Relevant features, such as word frequency or handwriting characteristics, are extracted from the preprocessed text. Machine learning models, including natural language processing (NLP) or computer vision

algorithms, analyze these features to evaluate the answers. The module generates reports summarizing evaluation outcomes for educators and integrates with existing systems for easy access to results. Continuous improvement is achieved by using feedback to refine machine learning models over time, ensuring more accurate and consistent evaluations.

3.4 MARK EVALUATE MODULE

A mark evaluation module for automated answer evaluation using machine learning is a sophisticated system that encompasses a series of intricate steps. It begins with the thorough preparation of data, involving the collection and preprocessing of answer sheets to ensure accuracy and consistency. Relevant features are then extracted from the preprocessed data to provide meaningful input for the machine learning models. These models are trained on labeled data, allowing them to learn and understand patterns in student responses.

Once trained, the machine learning models can efficiently evaluate answers and assign marks or scores automatically, eliminating the need for manual grading. This automation streamlines the evaluation process, saving time and resources for educators while ensuring fair and consistent grading practices. Moreover, by leveraging machine learning algorithms, the system can adapt and improve over time, enhancing its accuracy and reliability.

Implementing such a module not only improves efficiency but also enhances the overall quality of education. Students benefit from timely feedback and fair assessments, while educators can focus on more value-added tasks. Additionally, the system promotes transparency in grading, as the evaluation process is based on objective criteria. Overall, the mark evaluation module revolutionizes the assessment process in educational institutions, paving the way for more effective and equitable learning experiences.

CHAPTER-4

SYSTEM DESIGN

4.1 ARCHITECTURE DESIGN

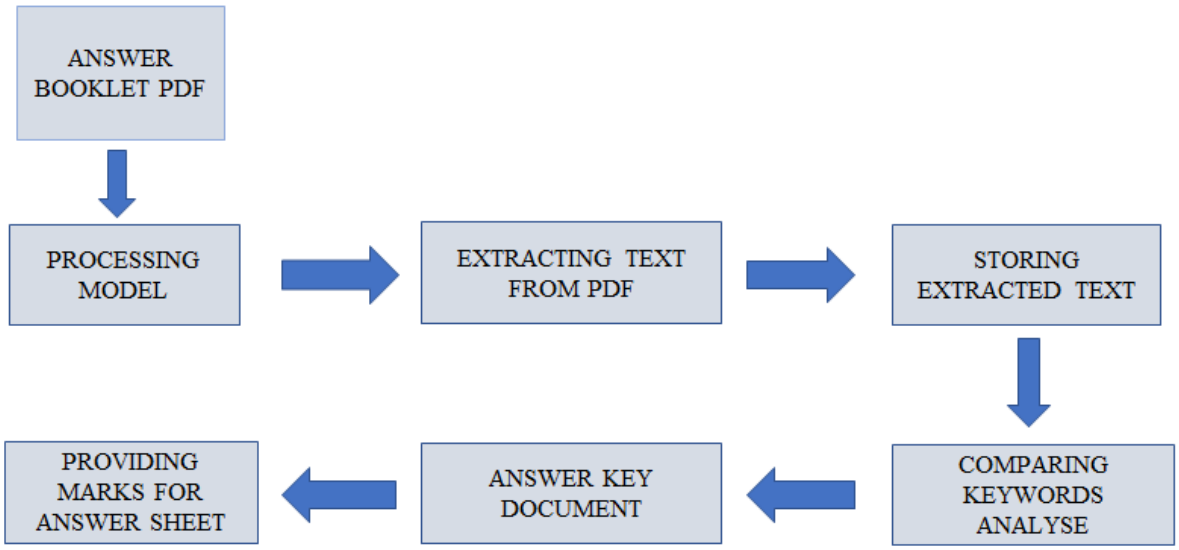

Figure.4.1 ARCHIETECTURE DESIGN

4.2 AUTOMATED ANSWER EVALUVATION

Automated answer evaluation using machine learning refers to the development and implementation of a system or project that utilizes machine learning techniques to assess the quality, correctness, or relevance of responses provided by individuals in various contexts. These contexts include educational assessments, customer support interactions, technical troubleshooting, or other domains where automated evaluation of answers is needed. By employing machine learning algorithms, such systems can analyze and interpret responses, providing valuable insights and feedback efficiently and accurately. This approach not only saves time and resources but also ensures consistency and fairness in evaluations across different scenarios.

The project typically involves:

- **Data Collection**: Gathering a dataset of questions or prompts along with corresponding correct or labeled answers.
- **Data Annotation**: Annotating the dataset by labeling the answers with scores, grades, or other evaluation metrics to serve as training data for machine learning models.
- **Feature Extraction**: Extracting relevant features from the answers, such as word frequency, syntax, semantic similarity, or other linguistic features, to represent the data in a format suitable for machine learning algorithms.
- **Model Training**: Training machine learning models, such as classification, regression, or natural language processing models, on the annotated dataset to learn patterns and relationships between the features of the answers and their corresponding evaluations.
- **Evaluation**: Evaluating the performance of the trained models on a separate validation dataset to assess their accuracy, precision, recall, or other evaluation metrics.
- **Deployment**: Integrating the trained models into a system or application that can automatically evaluate answers in real-time or batch processing mode, providing feedback or scores to users based on the evaluation results.

Automated answer evaluation using machine learning projects aim to streamline the evaluation process, reduce manual effort, improve consistency, and provide timely feedback to users or stakeholders in various domains.

4.3 GUI INPUT DESIGN

4.3.1 SYSTEM DESIGN

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built.

Once system requirements have been specified and analyzed, system design becomes the first of the three technical activities - design, code, and test - required to build and verify software. The importance of system design can be encapsulated in a single word: "Quality." Design is the stage where quality is fostered in software development, providing representations of software that can be assessed for quality. It is the only means through which we can accurately translate a customer's vision into a finished software product or system.

Software design serves as the foundation for all subsequent software engineering steps. Without a robust design, there is a risk of building an unstable system - one that will be difficult to test and whose quality cannot be assessed until the final stages of development.

During the design phase, progressive refinement of data structure, program structure, and procedural details takes place, which are then reviewed and documented. System design can be viewed from either a technical or project management perspective. From a technical standpoint, design encompasses four key activities - architectural design, data structure design, interface design, and procedural design.

These activities collectively ensure that the resulting software meets both technical and functional requirements while also facilitating future maintainability and scalability.

CHAPTER 5

SYSTEM DEVELOPMENT

5.1 SOFTWARE DESCRIPTION

5.2 HISTORY OF PYTHON

Python is a widely used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

5.3 PYTHON INVENTION

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application-based work in December of 1989 at Centrum Wiskunde &Informatica (CWI) which is situated in the Netherlands. It was started as a hobby project because he was looking for an interesting project to keep him occupied during Christmas**.**

The programming language in which Python is said to have succeeded is ABC Programming Language, which had interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped create ABC earlier in his career and had seen some issues with ABC but liked most of the features. After that what he did was very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and created a good scripting language that had removed all the flaws. The inspiration for the name came from the BBC's TV Show – 'Monty Python's Flying Circus', as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention

and hence he named it Python! He was the “Benevolent dictator for life” (BDFL) until he stepped down from the position as the leader on 12th July 2018. For quite some time he used to work for Google, but currently, he is working at Dropbox.

The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released, it had more than enough capability to provide classes with inheritance, several core data types of exception handling and functions.

Python has been an inspiration for many other coding languages such as Ruby, Cobra, Boo, Coffee Script ECMAScript, Groovy, Swift Go, OCaml, Julia, etc.

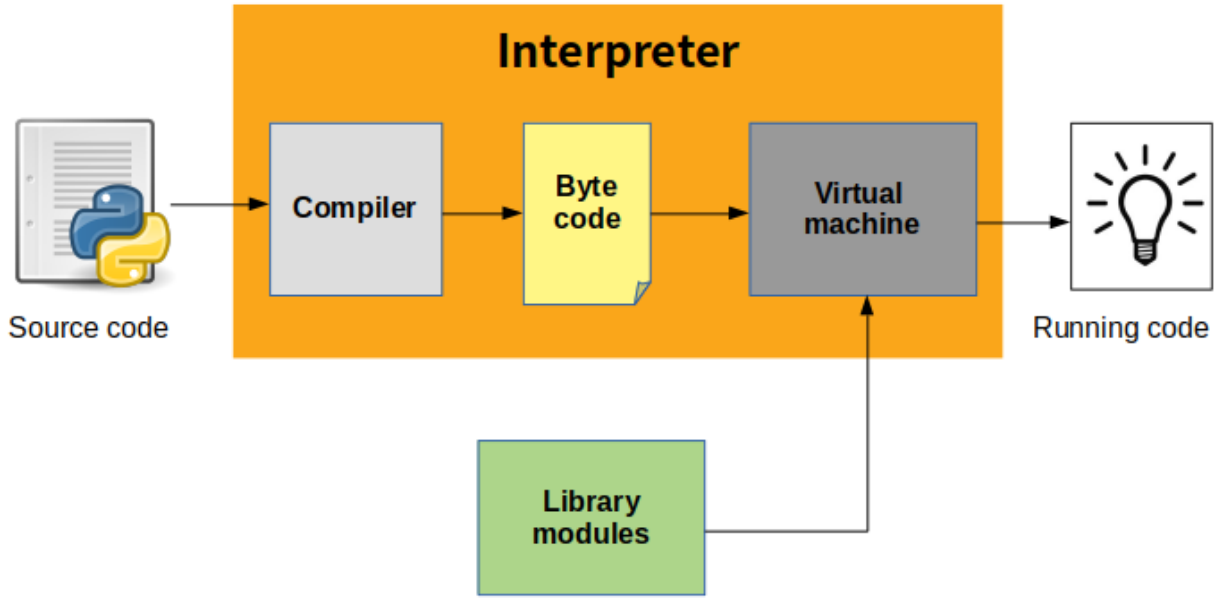

Figure. 5.1 PYTHON WORKFLOW

5.4 WORKING OF PYTHON LIBRARY

As is stated above, a Python library is simply a collection of codes or modules of codes that we can use in a program for specific operations. We use libraries so that we don't need to write the code again in our program that is already available. But how it works. Actually, in the MS Windows environment, the library files have a DLL extension (Dynamic Load Libraries). When we link a library with our program and run that program, the linker automatically searches for that library. It extracts the functionalities of that library and interprets the program accordingly. That's how we use the methods of a library in our program. We will see further, how we bring in the libraries in our Python programs.

5.5 PYTHON STANDARD LIBRARY

The Python Standard Library contains the exact syntax, semantics, and tokens of Python. It contains built-in modules that provide access to basic system functionality like I/O and some other core modules. Most of the Python Libraries are written in the C programming language. The Python standard library consists of more than 200 core modules. All these works together to make Python a high-level programming language. Python Standard Library plays a very important role. Without it, the programmers can't have access to the functionalities of Python. But other than this, there are several other libraries in Python that make a programmer's life easier. Let's have a look at some of the commonly used libraries:

1. **TensorFlow:** This library was developed by Google in collaboration with the Brain Team. It is an open-source library used for high-level computations. It is also used in machine learning and deep learning algorithms. It contains a large number of tensor operations. Researchers also use this Python library to solve complex computations in Mathematics and Physics.

2. Matplotlib: This library is responsible for plotting numerical data. And that's why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc.

3. Pandas: Pandas are an important library for data scientists. It is an open-source machine learning library that provides flexible high-level data structures and a variety of analysis tools. It eases data analysis, data manipulation, and cleaning of data. Pandas support operations like Sorting, Re-indexing, Iteration, Concatenation, Conversion of data, Visualizations, Aggregations, etc.

4. Numpy: The name "Numpy" stands for "Numerical Python". It is the commonly used library. It is a popular machine learning library that supports large matrices and multi-dimensional data. It consists of in-built mathematical functions for easy computations. Even libraries like TensorFlow use Numpy internally to perform several operations on tensors. Array Interface is one of the key features of this library.

5. SciPy: The name "SciPy" stands for "Scientific Python". It is an open-source library used for high-level scientific computations. This library is built over an extension of Numpy. It works with Numpy to handle complex computations. While Numpy allows sorting and indexing of array data, the numerical data code is stored in SciPy. It is also widely used by application developers and engineers.

6. Scrapy: It is an open-source library that is used for extracting data from websites. It provides very fast web crawling and high-level screen scraping. It can also be used for data mining and automated testing of data.

7. Scikit-learn: It is a famous Python library to work with complex data. Scikit-learn is an open-source library that supports machine learning. It supports variously supervised and unsupervised algorithms like linear regression, classification, clustering, etc. This library works in association with Numpy and SciPy.

8. PyGame: This library provides an easy interface to the Standard Directmedia Library (SDL) platform-independent graphics, audio, and input libraries. It is used for developing video games using computer graphics and audio libraries along with Python programming language.

9. PyTorch: PyTorch is the largest machine learning library that optimizes tensor computations. It has rich APIs to perform tensor computations with strong GPU acceleration. It also helps to solve application issues related to neural networks.

10. PyBrain: The name "PyBrain" stands for Python Based Reinforcement Learning, Artificial Intelligence, and Neural Networks library. It is an open-source library built for beginners in the field of Machine Learning. It provides fast and easy-to-use algorithms for machine learning tasks. It is so flexible and easily understandable and that's why is really helpful for developers that are new in research fields.

There are many more libraries in Python. We can use a suitable library for our purposes.

5.7 DESIGN PHILOSOPHY AND FEATURES

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of their features support functional programming and aspect-oriented programming (including metaprogramming and metaobjects). Many other paradigms are supported via extensions, including design by contract and logic programming.

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

Its design offers some support for functional programming in the Lisp tradition. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Its core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

Rather than building all of its functionality into its core, Python was designed to be highly extensible via modules. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, wrote: "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C; or use PyPy,

a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

A common neologism in the Python community is pythonic, which has a wide range of meanings related to program style. "Pythonic" code may use Python idioms well, be natural or show fluency in the language, or conform with Python's minimalist philosophy and emphasis on readability. Code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic.

5.8 SYNTAX AND SEMANTICS

Python is meant to be an easily readable language. Its formatting is visually uncluttered and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents its semantic structure. This feature is sometimes termed the off-side rule. Some other languages use indentation this way; but in most, indentation has no semantic meaning. The recommended indent size is four spaces.

Python does not support tail call optimization or first-class continuations, and, according to Van Rossum, it never will. However, better support for coroutine-like functionality is provided by extending Python's generators. Before 2.5, generators were lazy iterators; data was passed unidirectionally out of the generator. From Python 2.5 on, it is possible to pass data back into a generator function; and from version 3.3, it can be passed through multiple stack levels.

5.9 DEVELOPMENT

Python's development is conducted largely through the Python Enhancement Proposal (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues, and documenting Python design decisions. Python coding style is covered in PEP 8 Outstanding PEPs are reviewed and commented on by the Python community and the steering council.

Enhancement of the language corresponds with the development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues were originally discussed in the Roundup bug tracker hosted at by the foundation. In 2022, all issues and discussions were migrated to GitHub. Development originally took place on a self-hosted source-code repository running Mercurial, until Python moved to GitHub in January 2017.

CPython's public releases come in three types, distinguished by which part of the version number is incremented:

- Backward-incompatible versions, where code is expected to break and needs to be manually ported. The first part of the version number is incremented. These releases happen infrequently—version 3.0 was released 8 years after 2.0. According to Guido van Rossum, a version 4.0 is very unlikely to ever happen. Major or "feature" releases are largely compatible with the previous version but introduce new features. The second part of the version number is incremented. Starting with Python 3.9, these releases are expected to happen annually. Each major version is supported by bug fixes for several years after its release.
- Bugfix releases, which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release. Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented.

Many alpha, beta, and release-candidates are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large unit test suite during development.

The major academic conference on Python is PyCon. There are also special Python mentoring programs, such as Pyladies.

Python 3.10 deprecated wstr (to be removed in Python 3.12; meaning Python extensions need to be modified by then), and added pattern matching to the language.

5.9.1 API documentation generators

Tools that can generate documentation for Python API include pydoc (available as part of the standard library), Sphinx, Pdoc and its forks, Doxygen and Graphviz, among others.

5.9.2 Naming

Python's name is derived from the British comedy group Monty Python, whom Python creator Guido van Rossum enjoyed while developing the language. Monty Python references appear frequently in Python code and culture; for example, the metasyntactic variables often used in Python literature are spam and eggs instead of the traditional foo and bar. The official Python documentation also contains various references to Monty Python routines. The prefix Py- is used to show that something is related to Python.

5.10 OPEN CV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

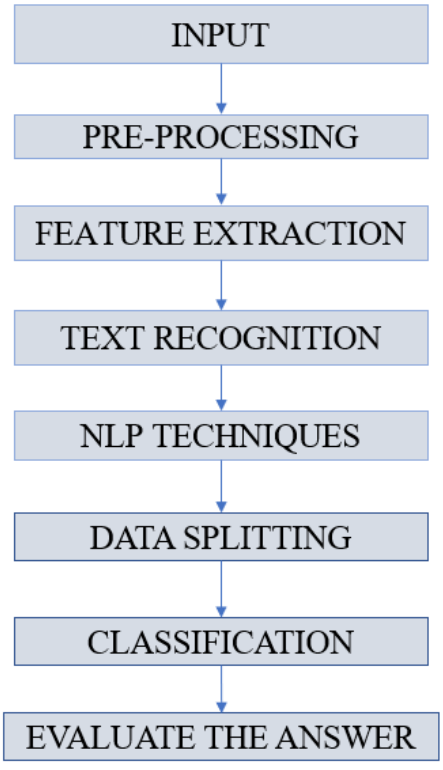

Figure. 5.2 FLOWCHART

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms.

5.11 OPENCV LIBRARY MODULES

Following are the main library modules of the OpenCV library.

5.11.1 Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array Mat, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name org.opencv.core.

5.11.2 Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.imgproc.

5.11.3 Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name org.opencv.video.

5.11.4 Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name org.opencv.videoio.

5.11.5 calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name org.opencv.calib3d.

5.11.6 features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name org.opencv.features2d.

5.11.7 Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.objdetect.

5.11.8 Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, org.opencv.imgcodecs and org.opencv.videoio.

5.12 PILLOW

Digital Image processing means processing the image digitally with the help of a computer. Using image processing we can perform operations like enhancing the image, blurring the image, extracting text from images, and many more operations. There are various ways to process images digitally. Here we will discuss the Pillow module of Python.

The Pillow module provides the open() and show() function to read and display the image respectively. For displaying the image Pillow first converts the image to a .png format (on Windows OS) and stores it in a temporary buffer and then displays it. Therefore, due to the conversion of the image format to .png some properties of the original image file format might be lost (like animation). Therefore, it is advised to use this method only for test purposes.

5.13 TENSORFLOW

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML, and gives developers the ability to easily build and deploy ML-powered applications.

TensorFlow provides a collection of workflows with intuitive, high-level APIs for both beginners and experts to create machine learning models in numerous languages. Developers have the option to deploy models on a number of platforms such as on servers, in the cloud, on mobile and edge devices, in browsers, and on many other JavaScript platforms. This enables developers to go from model building and training to deployment much more easily.

5.13.1 Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

5.13.2 Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

5.13.3 Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

- **TensorFlow** is basically a software library for numerical computation using **data flow graphs** where:
- **Nodes** in the graph represent mathematical operations.
- **Edges** in the graph represent the multidimensional data arrays (called **tensors**) communicated between them. (Please note that **tensor** is the central unit of data in TensorFlow).

5.14 TORCH

Deep Learning is a branch of Machine Learning where algorithms are written which mimic the functioning of a human brain. The most commonly used libraries in deep learning are Tensorflow and PyTorch. As there are various deep learning frameworks available, one might wonder when to use PyTorch. Here are reasons why one might prefer using Pytorch for specific tasks.

Pytorch is an open-source deep learning framework available with a Python and C++ interface. Pytorch resides inside the torch module. In PyTorch, the data that has to be processed is input in the form of a tensor.

CHAPTER-6

SYSTEM TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of software specification, design and coding. The increasing visibility of software as a system element and the attendant “costs” associated with a software failure are motivating forces for conference management system project well planned, thorough testing. It is not unusual for conference management system project software. Development organization to expend 40 percent of total project effort on testing. Hence the importance of software testing and its implications with respect to software quality cannot be overemphasized. Different types of testing have been carried out for conference management system project this system, and they are briefly explained below.

6.1 TYPES OF TESTS

Testing is the process of trying to discover every conceivable fault or weakness in a work product. The different types of testing are given below:

6.1.1 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 on knowledge 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 contains clearly defined inputs and expected results.

6.1.2 INTEGRATION TESTING

Integration testing plays a crucial role in software development, as it tests integrated software components to ensure they function effectively as a cohesive program. This type of testing is event-driven and primarily concerned with the basic outcomes of screens or fields. Despite individual components passing unit testing, integration tests verify that their combination is correct and consistent.

The main objective of integration testing is to identify and address any issues that arise from the interaction between these components, ensuring the overall integrity of the software system.

6.1.3 VALIDATION TESTING

Tests were conducted meticulously to ensure strict conformity with the specified requirements. Plans and procedures were meticulously crafted to guarantee the satisfaction of all functional requirements, leaving no stone unturned.

During the alpha testing phase, the software underwent rigorous scrutiny to achieve two main goals in preparing test plans. Firstly, a detailed test plan demonstrates a thorough understanding of program specifications, ensuring that no aspect is overlooked. Secondly, the test plan acts as a comprehensive tool during program testing, enabling thorough validation of the program's correctness and functionality.

This meticulous process ensures that the software not only meets the specified requirements but also operates flawlessly as intended, delivering optimal performance and reliability.

CHAPTER – 7

SOURCE CODE AND IMPLEMENTATION

main.py

```
import os
import tkinter as tk
from tkinter import *
from tkinter import messagebox, filedialog, PhotoImage
import mysql.connector
import PyPDF2

root = Tk()
root.title("College Management - Server")
root.geometry("1250x650")
root.config(bg="#1f97b5")

answer_pdf = open(r'E:/Arunai_CSE_Projects/Automatic Answer Evaluation/Answer_PDF.pdf',
mode='rb')

allocated_mark = 0
pdfdoc = PyPDF2.PdfReader(answer_pdf)
total_pages = len(pdfdoc.pages)

passshow_icon = PhotoImage(file='Images/eye.png')
passhide_icon = PhotoImage(file='Images/hide.png')
upload_icon = PhotoImage(file='Images/icon.png')
a_img = PhotoImage(file="Images/analyse.png")
process_icon = PhotoImage(file='Images/process.png')

password_mode = True
mark = 0
result = 0

try:
    login_frame = tk.Frame(root, width=350, height=335, bg="#1f97b5")
    signup_frame = tk.Frame(root, width=350, height=420, bg="#1f97b5")

    db = mysql.connector.connect(host='localhost', port=3306, database='college_details', user='root',
    password='')
    cursor = db.cursor()

    def change_signin():

        signup_frame.place_forget()
        login_frame.place(x=730, y=150)
```

```
    login_head = tk.Label(login_frame, text='College Login', bg="#1f97b5", fg='white',
                          font=('Microsoft YaHei UI Light', 23, 'bold'))
    login_head.place(x=70, y=10)

    def signin():
        username = login_user_entry.get()
        password = login_pass_entry.get()

        if username == 'Username' or password == 'Password':
            messagebox.showwarning("Warning", "Please enter missing Username or Password !!!")
        elif username == 'Vijay' and password == '1234':
            try:
                # db = mysql.connector.connect(host='localhost',
                #                              port=3306,
                #                              database='evaluate_details',
                #                              user='root',
                #                              password='')
                # cursor = db.cursor()

                messagebox.showinfo("Success", "Login Successfully ...")

                login_img.place_forget()
                login_frame.place_forget()
                login_title.place_forget()

                def analyse_pdf():
                    global filename
                    name = student_name.get()
                    reg = student_reg.get()
                    subject = student_subject.get()
                    dept = student_dept.get()
                    deg = student_deg.get()
                    college = student_college.get()

                    if name == 'Student Name' or reg == 'Student Register No' or subject == 'Student Subject'
or dept == 'Student Department' or deg == 'Student Degree' or college == 'Student College':
                        messagebox.showwarning("Warning", "Please enter student details to proceed ..")
                    else:
                        for p in range(0, total_pages):
                            pdf_page = pdfdoc.pages[p]
                            pdf_page = pdf_page.extract_text().split('. ')
                            for a in range(0, len(pdf_page)):
                                with open('Answer_Key.txt', 'r') as f:
                                    if pdf_page[a] in f.readline():
                                        allocated_mark += 1
                        mark = allocated_mark
```

```
        if mark > 35:
            result = 'P'
        else:
            result = 'F'

        cursor.execute("insert into evaluate_content values(%s,%s,%s,%s,%s,%s,%s,%s)",
                       (name, reg, subject, dept, deg, college, mark, result))
        db.commit()

        student_name.delete('0', 'end')
        student_name.insert('0', 'Student Name')
        student_reg.delete('0', 'end')
        student_reg.insert('0', 'Student Register No')
        student_subject.delete('0', 'end')
        student_subject.insert('0', 'Student Subject')
        student_dept.delete('0', 'end')
        student_dept.insert('0', 'Student Department')
        student_deg.delete('0', 'end')
        student_deg.insert('0', 'Student Degree')
        student_college.delete('0', 'end')
        student_college.insert('0', 'Student College')

        score_lable = Label(root, text="Score => " + str(mark),
                            font=('Microsoft YaHei UI Light', 16, 'bold'))
        score_lable.config(bg='#1f97b5', fg='white')
        score_lable.place(x=540, y=510)
        emoji = PhotoImage(file="Images/poor.png")
        emj_code = Label(root, image=emoji, bg='#1f97b5')
        emj_code.place(x=560, y=540)

        messagebox.showinfo("Success", "Paper Evaluated and Result stored in Database ..")

        score_lable.place_forget()
        emj_code.place_forget()

def upload_pdf():
    global filename
    filename = filedialog.askopenfilename(initialdir=os.getcwd(),
                                          filetype=(('PDF Files', '.pdf'), ('All Files', '*.*')))

project_title = Label(root, text="Automatic Answer Evaluation System")
project_title.config(font=('Microsoft YaHei UI Light', 24, 'bold'), bg='#1f97b5', fg='orange')
project_title.place(x=70, y=25)

analyse_img = Label(root, image=a_img, bd=0, bg='#1f97b5')
analyse_img.place(x=630, y=65)
```

```
        upload_label = Label(root, text="Upload Answersheet PDF format here")
        upload_label.config(bg='#1f97b5', fg='white', font=('Microsoft YaHei UI Light', 12))
        upload_label.place(x=180, y=180)

        upload_doc = Button(root, image=upload_icon, bd=0, bg='#1f97b5',
activebackground='#1f97b5',
                            fg='white')
        upload_doc.config(command=upload_pdf)
        upload_doc.place(x=280, y=90)

        def name_enter(e):
            student_name.delete('0', 'end')

        def name_leave(e):
            name = student_name.get()
            if name == '':
                student_name.insert('0', "Student Name")

        student_name = Entry(root, width=32, font=('Microsoft YaHei UI Light', 12))
        student_name.insert('0', "Student Name")
        student_name.place(x=178, y=240)
        student_name.bind('<FocusIn>', name_enter)
        student_name.bind('<FocusOut>', name_leave)

        def reg_enter(e):
            student_reg.delete('0', 'end')

        def reg_leave(e):
            reg = student_reg.get()
            if reg == '':
                student_reg.insert('0', "Student Register No")

        student_reg = Entry(root, width=32, font=('Microsoft YaHei UI Light', 12))
        student_reg.insert('0', "Student Register No")
        student_reg.place(x=178, y=290)
        student_reg.bind('<FocusIn>', reg_enter)
        student_reg.bind('<FocusOut>', reg_leave)

        def subject_enter(e):
            student_subject.delete('0', 'end')

        def subject_leave(e):
            subject = student_subject.get()
            if subject == '':
                student_subject.insert('0', "Student Subject")

        student_subject = Entry(root, width=32, font=('Microsoft YaHei UI Light', 12))
```

```
student_subject.insert('0', "Student Subject")
student_subject.place(x=178, y=340)
student_subject.bind('<FocusIn>', subject_enter)
student_subject.bind('<FocusOut>', subject_leave)

def dept_enter(e):
    student_dept.delete('0', 'end')

def dept_leave(e):
    dept = student_dept.get()
    if dept == '':
        student_dept.insert('0', "Student Department")

student_dept = Entry(root, width=32, font=('Microsoft YaHei UI Light', 12))
student_dept.insert('0', "Student Department")
student_dept.place(x=178, y=390)
student_dept.bind('<FocusIn>', dept_enter)
student_dept.bind('<FocusOut>', dept_leave)

def deg_enter(e):
    student_deg.delete('0', 'end')

def deg_leave(e):
    deg = student_deg.get()
    if deg == '':
        student_deg.insert('0', "Student Degree")

student_deg = Entry(root, width=32, font=('Microsoft YaHei UI Light', 12))
student_deg.insert('0', "Student Degree")
student_deg.place(x=178, y=440)
student_deg.bind('<FocusIn>', deg_enter)
student_deg.bind('<FocusOut>', deg_leave)

def college_enter(e):
    student_college.delete('0', 'end')

def college_leave(e):
    college = student_college.get()
    if college == '':
        student_college.insert('0', "Student College")

student_college = Entry(root, width=32, font=('Microsoft YaHei UI Light', 12))
student_college.insert('0', "Student College")
student_college.place(x=178, y=490)
student_college.bind('<FocusIn>', college_enter)
student_college.bind('<FocusOut>', college_leave)
```

```
            process_doc = Button(root, image=process_icon, bd=0, bg='#1f97b5',
activebackground='#1f97b5')
            process_doc.config(command=analyse_pdf)
            process_doc.place(x=280, y=540)

        except:
            print("Can't connect to MySQL Server.. If not, Please connect with MySQL Database in
Xampp..")

    def user_on_enter(e):
        login_user_entry.delete(0, 'end')

    def user_on_leave(e):
        login_username = login_user_entry.get()
        if login_username == '':
            login_user_entry.insert(0, 'Username')

    login_user_entry = tk.Entry(login_frame, width=25, border=0, bg="#1f97b5", fg='white',
                        font=('Microsoft YaHei UI Light', 11))
    login_user_entry.place(x=30, y=80)
    login_user_entry.insert(0, 'Username')
    login_user_entry.bind('<FocusIn>', user_on_enter)
    login_user_entry.bind('<FocusOut>', user_on_leave)

    tk.Frame(login_frame, width=295, height=2, bg='black').place(x=25, y=107)

    def pass_on_enter(e):
        login_pass_entry.delete(0, 'end')
        login_pass_entry.config(show='*')
        login_pass_show.config(image=passhide_icon)

    def pass_on_leave(e):
        login_pass_entry.config(show='')
        login_pass_show.config(fg='#fff', image='')
        login_password = login_pass_entry.get()
        if login_password == '':
            login_pass_entry.insert(0, 'Password')

    def passshow():
        global password_mode

        if password_mode:
            login_pass_show.config(image=passshow_icon)
            login_pass_entry.config(show='')
            password_mode = False
        else:
            login_pass_show.config(image=passhide_icon)
```

```
            login_pass_entry.config(show='*')
            password_mode = True

        login_pass_entry = tk.Entry(login_frame, width=25, border=0, bg="#1f97b5", fg='white',
                                    font=('Microsoft YaHei UI Light', 11))
        login_pass_entry.place(x=30, y=150)
        login_pass_entry.insert(0, 'Password')
        login_pass_entry.bind('<FocusIn>', pass_on_enter)
        login_pass_entry.bind('<FocusOut>', pass_on_leave)

        login_pass_show = tk.Button(login_frame, image='', border=0, bg='#1f97b5',
activebackground='#1f97b5',
                                    command=passshow)
        login_pass_show.place(x=280, y=135, width=50, height=50)

        tk.Frame(login_frame, width=295, height=2, bg='black').place(x=25, y=177)

        login_login_btn = tk.Button(login_frame, width=42, pady=7, text='Sign in', bg='orange', fg='white',
border=0,
                                    command=signin)
        login_login_btn.place(x=25, y=225)

        signup_label = tk.Label(login_frame, text="Don't have an account?", bg="#1f97b5", fg='white',
                                font=('Microsoft YaHei UI Light', 9))
        signup_label.place(x=75, y=280)

        signup_btn = tk.Button(login_frame, width=6, text='Sign Up', border=0, bg="#1f97b5", fg='white',
cursor='hand2',
                               activebackground='#1f97b5', activeforeground='white', command=change_signup)
        signup_btn.place(x=215, y=280)


    def change_signup():
        login_frame.place_forget()
        signup_frame.place(x=730, y=150)

        login_head = tk.Label(signup_frame, text='College Register', bg="#1f97b5", fg='white',
                              font=('Microsoft YaHei UI Light', 23, 'bold'))
        login_head.place(x=50, y=10)

        def signup():
            username = register_user_entry.get()
            password = register_pass_entry.get()
            college = register_college_entry.get()
            univ = register_univ_entry.get()

            if username == 'Username' or password == 'Password' or college == 'College Name' or univ ==
```

```
'University Name':
            messagebox.showwarning("Warning", "Please enter missing elements to complete registration
!!!")
        else:
            # cursor.execute("insert into college_content values(%s,%s,%s,%s)", (username, college, univ,
password))
            # db.commit()
            messagebox.showinfo("Success", "Registered Successfully!!")

    def user_on_enter(e):
        register_user_entry.delete(0, 'end')

    def user_on_leave(e):
        register_username = register_user_entry.get()
        if register_username == '':
            register_user_entry.insert(0, 'Username')

    register_user_entry = tk.Entry(signup_frame, width=25, border=0, bg="#1f97b5", fg='white',
                                   font=('Microsoft YaHei UI Light', 11))
    register_user_entry.place(x=30, y=80)
    register_user_entry.insert(0, 'Username')
    register_user_entry.bind('<FocusIn>', user_on_enter)
    register_user_entry.bind('<FocusOut>', user_on_leave)

    tk.Frame(signup_frame, width=295, height=2, bg='black').place(x=25, y=110)

    def college_on_enter(e):
        register_college_entry.delete(0, 'end')

    def college_on_leave(e):
        register_college = register_college_entry.get()
        if register_college == '':
            register_college_entry.insert(0, 'College Name')

    register_college_entry = tk.Entry(signup_frame, width=25, border=0, bg="#1f97b5", fg='white',
                                      font=('Microsoft YaHei UI Light', 11))
    register_college_entry.place(x=30, y=140)
    register_college_entry.insert(0, 'College Name')
    register_college_entry.bind('<FocusIn>', college_on_enter)
    register_college_entry.bind('<FocusOut>', college_on_leave)
    tk.Frame(signup_frame, width=295, height=2, bg='black').place(x=25, y=170)

    def univ_on_enter(e):
        register_univ_entry.delete(0, 'end')

    def univ_on_leave(e):
        register_univ = register_univ_entry.get()
```

```
        if register_univ == '':
            register_univ_entry.insert(0, 'University Name')

    register_univ_entry = tk.Entry(signup_frame, width=25, border=0, bg="#1f97b5", fg='white',
                                   font=('Microsoft YaHei UI Light', 11))
    register_univ_entry.place(x=30, y=200)
    register_univ_entry.insert(0, 'University Name')
    register_univ_entry.bind('<FocusIn>', univ_on_enter)
    register_univ_entry.bind('<FocusOut>', univ_on_leave)

    tk.Frame(signup_frame, width=295, height=2, bg='black').place(x=25, y=230)

    def pass_on_enter(e):
        register_pass_entry.delete(0, 'end')
        register_pass_entry.config(show='*')
        register_pass_show.config(image=passhide_icon)

    def pass_on_leave(e):
        register_pass_entry.config(show='')
        register_pass_entry.insert(0, 'Password')
        register_pass_show.config(fg='#fff', image='')

    def passshow():
        global password_mode

        if password_mode:
            register_pass_show.config(image=passshow_icon)
            register_pass_entry.config(show='')
            password_mode = False
        else:
            register_pass_show.config(image=passhide_icon)
            register_pass_entry.config(show='*')
            password_mode = True

    register_pass_entry = tk.Entry(signup_frame, width=25, border=0, bg="#1f97b5", fg='white',
                                   font=('Microsoft YaHei UI Light', 11))
    register_pass_entry.place(x=30, y=260)
    register_pass_entry.insert(0, 'Password')
    register_pass_entry.bind('<FocusIn>', pass_on_enter)
    register_pass_entry.bind('<FocusOut>', pass_on_leave)

    register_pass_show = tk.Button(signup_frame, image='', border=0, bg='#1f97b5',
activebackground='#1f97b5',
                                   command=passshow)
    register_pass_show.place(x=280, y=250, width=50, height=50)
    tk.Frame(signup_frame, width=295, height=2, bg='black').place(x=25, y=290)
```

```
        signup_btn = tk.Button(signup_frame, width=37, pady=3, text='Sign up', bg='orange', fg='white',
border=0,
                               command=signup,
                               font=('Microsoft YaHei UI Light', 10))
        signup_btn.place(x=25, y=325)

        signup_label = tk.Label(signup_frame, text="Already have an account?", bg="#1f97b5", fg='white',
                                font=('Microsoft YaHei UI Light', 9))
        signup_label.place(x=65, y=375)

        signup_btn = tk.Button(signup_frame, width=6, text='Sign In', border=0, bg="#1f97b5", fg='white',
                               cursor='hand2',
                               activebackground='#1f97b5', activeforeground='white', command=change_signin)
        signup_btn.place(x=215, y=375)


    change_signin()

    login_title = Label(root, text="Automatic Answer Evaluation System")
    login_title.config(font=('Microsoft YaHei UI Light', 24, 'bold'), bg='#1f97b5', fg='orange')
    login_title.place(x=630, y=25)
    l_img = PhotoImage(file="Images/login.png")
    login_img = Label(root, image=l_img, bd=0, bg='#1f97b5')
    login_img.place(x=80, y=85)
    root.mainloop()

except:
    print("Can't connect to MySQL Server.. If not, Please connect with MySQL Database in Xampp..")
```

CHAPTER 8

RESULT AND OUTPUT

OUTPUT:

8.1 LOGIN PAGE

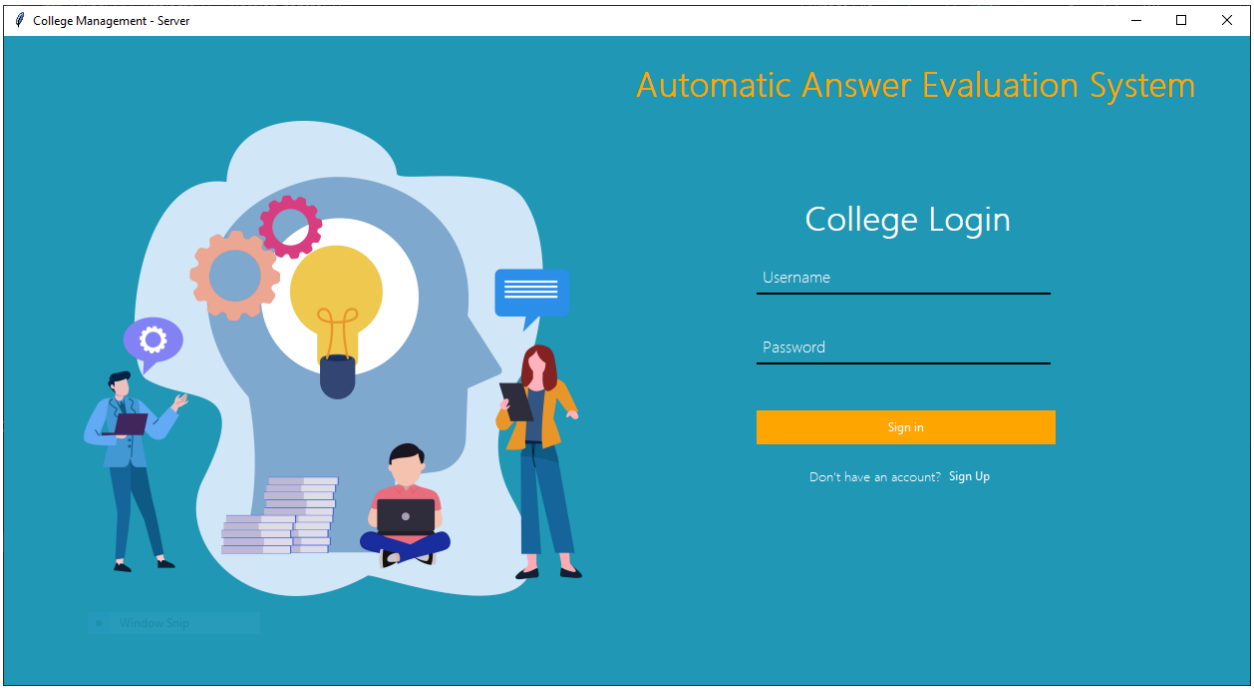

Figure. 8.1 LOGIN PAGE

The project also includes a web application where users can log in using their user ID and password. If there is no existing ID, they can create a new one by signing up. The login page prompts users to enter their user ID and password. Upon submission, the application verifies the credentials against the existing database. If the credentials are correct, users are directed to their respective dashboard. If the credentials are not found or incorrect, users are prompted to try again or sign up.

For the sign-up process, users are required to provide necessary details such as name, email, password, etc. Once the details are submitted, the application checks if the user ID is unique and stores the new user information in the database.

8.2 COLLEGE REGISTER

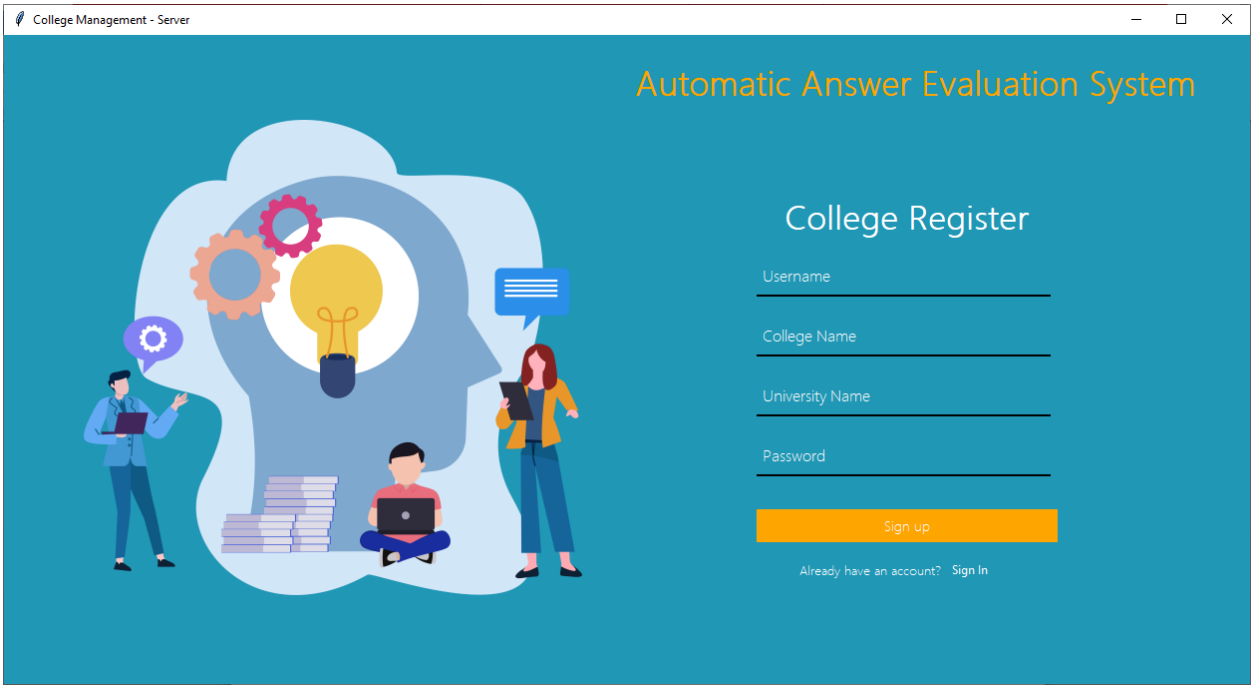

Figure. 8.2 COLLEGE REGISTRATION

The registration process for college management involves providing necessary details such as college name, address, contact information, etc. Once the details are submitted, the application checks if the college name is unique and stores the information in the database. After successful registration, college management receives login credentials.

8.3 LOGIN SUCCESSFULL

Figure 8.3 LOGIN SUCCESSFULL

After entering the login ID and password, if they are correct, a successful login message will be displayed to the user.

8.4 UPLOAD STUDENT DATA

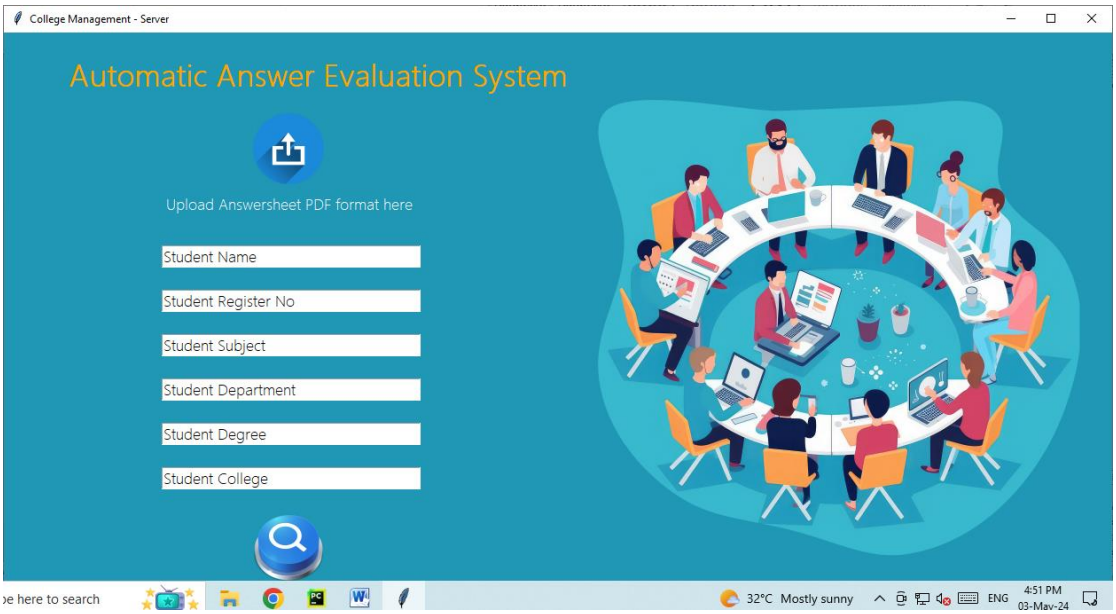

Figure. 8.4 UPLOAD STUDENT DATA

The student enters their name, registration number, department, and subject name into the provided fields. They then upload their answer booklet PDF and click the submit button. Upon submission, the system processes the data and uploaded file.

8.5 EVALUATED SUCCESSFULL

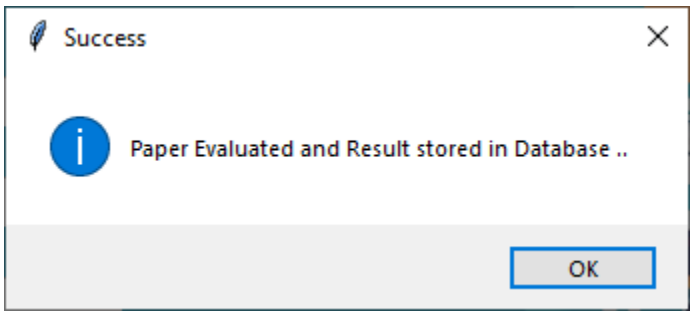

Figure. 8.5 EVALUATED SUCCESSFUL

If successful, a confirmation message is displayed, indicating that the answer booklet has been successfully submitted.

8.6 STUDENTS DATABASE

NAME	REG	SUBJECT	DEPT	DEG	COLLEGE	MARK	RESULT
Aaship	510420104001	Python	CSE	BE	Arunai Engineering College	27	F
CHANDRU	510420104020	PYTHON	CSE	BE	Arunai Engineering College	65	P
Ajay	510420104009	Python	CSE	BE	Arunai Engineering College	71	P
Saranya	510420104050	Python	CSE	BE	Arunai Engineering College	91	P
arun	510420104020	python	CSE	BE	Arunai Engieerinf College	18	F
ashok	510420104015	python	CSE	BE	Arunai Engineering College	83	P

1/1

Figure. 8.6 STUDENTS DATABASE

The web application displays the database of student marks and details. This database includes information such as student names, registration numbers, departments, subjects, and corresponding marks. Users, such as administrators or instructors, can access this database to view and manage student records, track academic progress, and generate reports.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

In this paper introduces an algorithm developed for the evaluation of theoretical answers, offering a method that assigns marks based on keyword matching. This approach significantly reduces manual workload and accelerates the result evaluation process. The procedure begins with the collection of the answer copy from the student, which is then scanned.

Subsequently, the algorithm processes the scanned image, considering various factors such as answer length, inclusion of definitions, and coverage of important keywords specified by the teacher. By automating this process, the algorithm not only streamlines evaluation but also saves considerable time. Moreover, it ensures consistent and efficient grading, benefiting both educators and students alike.

9.2 FUTURE WORK

This model has the potential to be trained for different languages spoken across India. By collecting datasets of various handwritten languages, the system can evaluate answers written in languages other than English. Additionally, the system can accurately evaluate overwritten alphabets and other words with precision. This capability ensures that regardless of the language or writing style used by the student, the system can provide accurate evaluations consistently.

CHAPTER 10
REFERENCES

[1] Nandini, P. Uma Maheswari, "Automatic assessment of descriptive answers in online examination system using semantic relational features", The Journal of Supercomputing, 2023.

[2]AditiTulaskar, AishwaryaThengal, KamleshKoyande, "Subjective Answer Evaluation System", Volume 7 Issue No.4 International Journal of Engineering Science and Computing, Pp.10457-10459, 2022.

[3] Vimal P. Parmar, C K Kumbharana, "Analysis Of Different Examination Patterns Having Question Answer Formulation, Evaluation Techniques And Comparison Of MCQ Type With One Word Answer For Automated Online Examination" Interna- tional Journal of Scientific and Research Publications, Volume 6, Issue 3, Pp.459- 463, 2021.

[4] asmithadhokrat, gitehanumant r, c. namrataMahender, Automated Answering For SubjectiveExamination", International Journal Of Computer Applications.VOL.56. Pp.14-17, 2021.

[5] PranaliNikam, MayuriShinde, RajashreeMahajan and ShashikalaKadam, Automatic Evaluation of Descriptive Answer Using Pattern Matching Algorithm", International Journal of Computer Sciences and Engineering, Pp.69-70, 2015.

[6] Ravikumar M, Sampath Kumar S and Shivakumar G PiyushPatil, Sachin Patil, VaibhavMiniyar, AmolBandal, "Subjective Answer Evaluation
Using Machine Learning", International Journal of Pure and Applied Mathematics, Pp.01-13, 2018.

[7] DhananjayKulkarni, Ankit Thakur, JitendraKshirsagar, Y. Ravi Raju4, "Auto-matic OMR Answer Sheet Evaluation using Efficient Reliable OCR System", In- ternational Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 3, Pp.688-690, 2017