“SMART SUBJECTIVE ANSWER ASSESSMENT SYSTEM USING MACHINE LEARNING”

A Dissertation Submitted to

P.E.S COLLEGE OF ENGINEERING MANDYA- 571 401

(An Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

In partial fulfillment of the requirement for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Submitted By

OMPRASAD M
(4PS21MC036)

Under the Guidance of

Internal Guide		External Guide
Dr H.R. DIVAKAR		**Mr. SANTHOSH**
Associate professor		Software Engineer
Department of MCA	v	Parvam Software Solutions
P.E.S.C.E Mandya		Bengaluru

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

P.E.S. College of Engineering, Mandya-571 401
2022-2023

P.E.S COLLEGE OF ENGINEERING
Mandya-571 401

(An Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that, **OMPRASAD M (4PS21MC036),** has satisfactorily completed the dissertation work report entitled **"Smart Subjective Answer Assessment System Using Machine Learning"** in partial fulfillment of the requirement for the award of the degree of **Master of Computer Applications,** PES College of Engineering, Mandya**,** Visvesvaraya Technological University, Belagavi during the year 2022-2023. It is certified that all corrections/suggestions indicated in Internal Assessment have been incorporated in the report. This dissertation report has been approved as it satisfies the academic requirements in respect of dissertation work prescribed for the degree in Master of Computer Applications.

Signature of the Internal Guide

Dr H.R. DIVAKAR
Associate Professor
Department of MCA
P.E.S.C.E., Mandya

Signature of the HOD

Dr. H.P MOHAN KUMAR
Professor & HOD
Department of MCA
P.E.S.C.E., Mandya

Signature of the Principal

Dr. R.M. Mahalinge Gowda
Principal, P.E.S.C.E, Mandya

Details of Dissertation work Viva Voce Examination Held			
SL. No	**Examiners**		**Date**
	Name	**Signature**	
1			
2			

ParvaM Software Solutions

To

The HOD

Department of Computer Science (MCA)

P.E.S College of Engineering

Mandya

This is to inform that Om Prasad .M (4PS21MC036) MCA Student of your institution is permitted to undergone project work in our organization under the guidance of our Technical team from 1-3-2023 to 15-7-2023.

The candidates will be associated with us and shall be entitled to avail our facility as per the company's rules and practices.

Regards

[illegible]

Santosh

Authorized Signatory

ParvaM Software Solutions

DECLARATION

I, **OMPRASAD M (4PS21MC036)**, hereby declare that the dissertation work entitled "**Smart Subjective Answer Assessment System Using Machine Learning**", has been independently carried out by me under the guidance of **Dr. H.R.Divakar,** Associate Professor Department of MCA, P.E.S.C.E, Mandya and **Mr. Santhosh**, Parvam Software Solutions Bengaluru, in partial fulfillment of the requirement of the degree Master of Computer Application (MCA), PES College of Engineering, Mandya an Autonomous Institute under VTU, Belgaum. I further declare that I have not submitted this dissertation either in part or in full to any other university for the award of any degree.

Date: **OMPRASAD M**
[4PS21MC036]

Place: Mandya

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all people who have helped in the successful completion of this dissertation work, above all I thank almighty for being there all the time when I needed his help.

I would like to express my heartfelt gratitude **to Dr. R.M. Mahalinge Gowda**, Principal, P.E.S College of Engineering, Mandya for help and rendered in undertaking the project.

I would like to express my immense gratitude to **Dr. H.P. Mohan Kumar**, Professor and HOD, Dept. of MCA, PES College of Engineering, Mandya for his valuable assistance and Co-operation.

It pays my immense respect to my internal guide **Dr. H.R. Divakar**, Associate Professor, Dept. Of MCA, PES College of Engineering, Mandya for giving support and valuable guidancethroughout my dissertation work.

It is pleasure to acknowledge my External guide **Mr**. **Santhosh**, Parvam Software Solutions, Bengaluru, for giving support and valuable guidance throughout my dissertation work.

It is also pleasure to express my gratitude to all **teaching and non- teaching** staff members of MCA department for their encouragement and providing valuable requirements.

OMPRASAD M
(4PS21MC036)

ABSTRACT

Subjective answer evaluation is an essential task that requires considerable time and effort. In this research, we propose a machine learning and natural language processing-based approach to evaluate subjective answers. The proposed system uses a Natural language processing along with Multinomial Naive Bayes (MNB) model to classify the subjective answers into different categories. The system's methodology involves data pre-processing, feature extraction, and classification. The objective of this research is to improve the accuracy and efficiency of subjective answer evaluation.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

ParvaM Software Solutions

PARVAM
NEXT IS NOW

TO WHOMSOEVER IT MAY CONCERN

This is to inform that Omprasad.M (4PS21MC036) MCA Student of PES College of Engineering, Mandya Successfully Completed project work in our organization under the guidance of our Technical team from 1-3-2023 to 15-7-2023.

The candidates be associated with us and entitled to avail our facility as per the company's rules and practices.

Regards

[illegible]

ParvaM Software Solutions

Santosh

Authorized Signatory

117, New Extension, Chikkabanavara, Hessaraghatta Main Road, Bangalore - 560090

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

When evaluating subjective remarks, teachers, professors, examiners, or recruiters may find this tool beneficial. In contrast to traditional assessment, the system can give an accurate and effective evaluation of subjective answers by relying on machine learning and natural language processing. For in-depth examinations or when a rapid analysis is necessary, this could be extremely useful.

1.2 MOTIVATION OF PROJECT

This study's goal is to propose an approach for judging subjective responses that is based on machine learning and natural language processing. The precise goals are to purge the subjective replies of noise and unimportant data.

1. To take the pertinent characteristics out of the data that has been pre-processed.
2. To develop a Multinomial Naive Bayes (MNB) model for classifying the illogical responses.
3. To assess how accurately the MNB model classified the arbitrary responses.
4. To assess how the suggested system performs in comparison to other assessment techniques.

Our Solution

Three processes make up the recommended system's methodology: feature extraction, data pre-processing, and classification.

1.3 COMPANY PROFILE

An Overview of the Company

The company's headquarters are located in a facility that spans 2500 square feet and houses all of its operations. The facility is well-equipped and provides an atmosphere that is beneficial for the development of software. The infrastructure includes the following features: Two training rooms, each with a capacity of 30 seats, which enables us to conduct complete training programmers. a specialized center for the development process that has enough seats for fifteen people and provides an excellent working environment for our team of developers. A continuous source of electricity provided by a 5-kilovolt uninterruptible power supply (UPS), which guarantees uninterrupted operations even when the power is out. The availability of filtered drinking water for the comfort and convenience of our staff. a pleasant working environment that is well-lit and ventilated, to welcome visitors and clients...an appealing greeting area to welcome visitors and clients...audio/video technology to promote effective communication and collaboration...well-lit and ventilated premises to create...

The Method We Use

By delivering services that go above and beyond, we hope to foster long-term partnerships with each of our customers that are mutually beneficial. Our method involves the following steps obtaining a comprehensive grasp of the business requirements and goals of the customer. Working directly with the customer to develop innovative software solutions in close collaboration. Developing applications that are both resilient and scalable by utilizing the most recent technology and industry best practices. Keeping an open and honest line of communication open with the customer throughout the entirety of the product development process. Meeting the schedules and staying within the allotted budget for the delivery of high-quality software solutions.

CHAPTER 2

LITERATURE SURVEY

2.1 REVIEW OF LITERATURE SURVEY

2.1.1 Subjective Answer Evaluation Using Machine Learning

These days, negative evaluation methods are used to assess subjective writing. Assessing the subjective responses has a significant role. Depending on one's feelings, a human's evaluation may be of varying quality. Every output in machine learning is based on the user's input data. To solve this problem, our suggested technique makes use of machine learning and NLP. To investigate the subjective response, our system performs operations such tokenizing words and phrases, part of speech tagging, chunking, chinking, lemmatizing words, and word netting.

2.1.2 Subjective Answers Evaluation Using Machine Learning and Natural Language Processing

Reviewing subjective articles manually is a tough and difficult undertaking. When employing artificial intelligence (AI) to assess subjective articles, a fundamental hurdle is a lack of comprehension and acceptance of the conclusions. Several attempts have been made to evaluate student responses using computer science. To do this, however, the majority of the task calls either precise language or standard counts. There aren't enough carefully selected data sets, too. The new methodology of this work makes use of many machine learning and natural language processing methods.

2.1.3 Online Subjective Answer Verifying System using Artificial Intelligence

University and board examinations are administered offline each year. Many students show up for subjective tests. Such a big quantity of articles needed a lot of manual work to evaluate. Sometimes, the evaluator's disposition will have an impact on the evaluation's quality. The evaluation process requires a lot of time and effort.

2.2 PROBLEM STATEMENT

The time-consuming and labor-intensive job of analyzing subjective replies requires the manual review and assessment of each response by human assessors. This method can be time-consuming and subjectivity-prone, especially when dealing with a high number of replies. The issue is the dearth of an automated system that can efficiently, correctly, and swiftly examine subjective responses.

2.3 EXISTING SYSTEM

Manual evaluation is frequently used to assess subjective responses. Reading the response and evaluating it in line with standards like application to the issue, coherence, clarity, and depth of knowledge are involved in this.

Following the evaluation, the evaluator would provide a score or grade, which can be subjective and impacted by their own biases or preferences. This procedure, particularly for lengthy evaluations, can be labor- and time-intensive and may lead to grade inconsistencies or inequalities.

Rubrics or scoring criteria are typically used to maintain uniformity across several assessors and standardize the assessment process, which increases the dependability of manual evaluation. These rubrics usually outline the criteria for evaluating subjective responses and provide in-depth explanations or examples of various performance levels.

2.4 PROPOSED SYSTEM

The suggested solution uses machine learning and natural language processing to analyze subjective responses in an efficient and objective manner. The technique of the system is composed of feature extraction, pre-processing of the data, and classification.

The suggested system starts with data pre-processing. To prepare the raw data for analysis, this technique encompasses cleaning and modification. In order to evaluate answers that were submitted subjectively, this may include deleting extraneous information, such as headers and footers, and converting the text into an analysis-ready format, such tokens or vectors. Pre-processing is an essential stage since it has a significant impact on how accurate the categorization is in the end.

Features extraction is the following stage in the suggested system. The goal of this stage is to find and remove pertinent features from the pre-processed data. The subjective evaluation of responses may take into account the coherence and structure of the response, the breadth and correctness of the information offered in the response, the use of acceptable language and grammar, and the relevance of the response to the question. The characteristics are subsequently converted into quantifiable quantities for study.

The suggested system's categorization step comes last. The subjective replies have now been split into a variety of categories using a machine learning technique. The Multinomial Naive Bayes (MNB) model is applied in this case for classification. MNB is a probabilistic approach that categorizes fresh samples in accordance with the frequency of qualities in the training data using the Bayes theorem. The MNB model is trained on a collection of labelled samples for subjective response evaluation in order to ascertain the link between the extracted variables and the various answer categories. In order to categories fresh responses using the qualities discovered from them, the trained model may be employed.

2.5 SOFTWARE DEVELOPMENT TOOLS

2.5.1 Python Language

Python is a highly regarded and widely used programming language that is put to use in a number of industries. Some of these sectors include web development, data analysis, artificial intelligence, and scientific computing, to mention a few. Python is also well regarded by its users. Because computer code is meant to be simple and easy, programmers have the ability to develop code that is both clear and concise in their inventions. This allows for more efficiency. Python is usually regarded to be an approachable programming language for beginners due to the fact that it places a strong focus on the readability of code, makes extensive use of indentation, and possesses a reasonably easy grammar.

5.1.2 Machine Learning

The objective of the branch of artificial intelligence (AI) known as machine learning is to develop models and algorithms that will allow computers to learn, make predictions, and make judgments without the need for explicit instructions. This will enable the computers to learn new information, make decisions, etc.

2.1.3 Python IDLE

Python IDLE is a piece of software that is included with the package that includes the Python programming language. This package may be downloaded from the Python website. It is also known by its other name, the Integrated Development and Learning Environment, which you might be familiar with. Users are supplied with a user interface that is easy to understand, which enables users to write Python programmers, execute those programmers, and debug those programmers using the software.

2.1.4 Flask Web kit Framework

Python is the programming language used to create Flask, which is a web framework that is well-known for its versatility as well as its lightweight nature. As a consequence of this, it is a well-liked option among developers due to the fact that it makes the construction of apps for the internet both quick and simple. Its development takes use of the Model– View–Controller (MVC) paradigm because of the benefits it brings in terms of being scalable and easy to maintain. Flask is well-known for its user-friendliness due to the fact that it has a straightforward design and a small codebase, both of which make it easy to learn and simple to put into practice. This has contributed to Flask's widespread popularity. It focuses a major emphasis on offering the core features necessary for web development by providing programmers with the ability to pick and choose which libraries and components to include in their code based on the particular requirements of their projects

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

The Software Requirement Specifications (SRS) document is a significant document that gathers and specifies the demands of a software system, both functional and non- functional needs. This document may also be abbreviated as SRS. Communication between all of the parties engaged in the software development process, including clients, developers, and testers, is essential to the project's success and should take place as often as possible. The major purpose of the SRS document is to achieve a level of consensus on the program's capabilities, limits, and general goals. This will allow for the document to be used effectively. The SRS document is burdened with the obligation of generating an agreement between all of the parties involved, including the client or customer and the development team. This agreement must be legally enforceable in order to be considered valid. When this is done, it ensures that all individuals who are taking part in the process of building software are completely aware about the specific requirements and objectives that need to be met in order for the process to be successful. This can be performed by ensuring that everyone involved is fully informed about the requirements and objectives.

3.2 SDLC METHODOLOGY

The Software Development Life Cycle, or SDLC for short, is the term used to describe the process or approach used to construct software systems. It offers a methodical way to direct the many stages of software development, from early design through ultimate deployment and maintenance.

Water Fall Model

The waterfall paradigm guides software development through a predetermined, sequential process. The software development process is divided into various stages, and each one builds on the results of the one before it. The waterfall model's phases typically include:

Requirements Gathering:

The project requirements are determined, outlined, and verified at this phase. This calls for gathering information from stakeholders, comprehending their demands, and establishing the project's scope. A full requirements specification document will contain a listof the criteria.

System Design:

The system design phase is focused with creating the architecture, parts, and user interfaces of the software system based on the requirements. It entails developing intricate system-level designs, which include requirements for modules, user interfaces, and database architectures.

Implementation:

Based on the design criteria, the software system is created at this step. Individual modules or components are implemented when all coding and programming responsibilities have been completed. The objective is to convert the design into functional code.

Testing:

Following implementation, testing is started. To ensure that the programmed satisfies the stated requirements and performs as intended, test cases and scenarios are conducted. User acceptance testing, system testing, integration testing, and unit testing are a few forms of testing.

Deployment:

After being completely tested and bug-fixed, the programmed is made available to the clients or end users. This phase entails setting up the required hardware and software infrastructure, as well as preparing the programmed for usage and production.

After a programmed has launched, bugs, errors, and improvements are addressed during the maintenance phase. Customer support, updates, and bug fixes are all part of it.

3.3 FUNCTIONAL REQUIREMENTS

A software system's functional requirements are necessary components that have a major influence on the system's behaviors, features, and functions. These requirements are sometimes referred to as functional specifications. They give an in-depth explanation of what the software should be able to perform and how it should respond to a range of different inputs and circumstances. These requirements give specific information on the interactions that should take place between the user and the system, as well as the results and replies that should be supplied by the system. To put it another way, functional requirements are a means of identifying the basic features and behaviors that software has to have in order for it to be able to fulfil the goals that it has set for itself.

1. **Training Coordinator Module**: This module is used by the Training Coordinator of the educational institution, who oversees the training phase of the application. The Training Coordinator can upload the training dataset, initiate the training process using the Multinomial Naive Bayes algorithm, create a model, and view the training results through graphical representations.

2. **Evaluation Officer Module**: This module is used by the Evaluation Officer of the educational institution, who oversees the evaluation phase of the application. The Evaluation Officer can upload the test dataset and view the test results.

3. **Testing Administrator Module**: This module is used by the Testing Administrator of the educational institution, who oversees the testing phase of the application. The Testing Administrator has to first register with his details, and overseeing the test data uploads. They can also check the answer scores and manage the testing process within the application.

3.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements (NFRs) define the qualities, characteristics, and constraints that a software system should possess. Unlike functional requirements that describe what the software should do, non-functional requirements specify how the software should behave or perform. They focus on aspects such as performance, reliability, security, usability, scalability, and other factors that contribute to the overall quality and user experience of the system. Here are some common categories of non-functional requirements: Here are some common categories of non-functional requirements:

Performance:

Using the performance requirements for a certain system, one is able to arrive at an estimate for the reaction times, throughput, and resource utilization that are expected to be associated with a particular system. It is necessary to take into consideration a number of factors, such as the rate at which users interact with the system, the efficiency with which the system processes information, the amount of time that passes during network communication, and the capability of the system to deal with increased quantities of work.

Reliability:

The criteria are necessary because they determine the extent to which a system is capable of reliably carrying out the duties for which it was created without encountering any faults. This makes the criteria crucial. This includes characteristics such as the system's availability, fault tolerance, error handling procedures, and the capacity to maintain data integrity. Also included is the availability of the system.

Security:

The phrase "requirements" refers to the specific safeguards and precautions that need to be taken in order to protect a system and its data from any unauthorized access, breaches, or malicious activities that may be carried out in the future. These safeguards and precautions are necessary in order to maintain the integrity of the system and its data. Some of the topics that are covered in this document include compliance with relevant security standards, authentication processes, authorization protocols, encryption methods and methodologies, and more. Other topics include encryption techniques and methodology.

Usability:

They are necessary because they ensure that the software system will be easy to use and that it will give users with a satisfactory experience. In this context, issues such user interfaces that are easy to use, accessibility features that are aimed towards users with disabilities, substantial documentation, and any necessary user training that may be required are all things that should be taken into account.

Scalability:

Refers to the system's capability to handle larger workloads or accommodate growing user demands. This refers to the ability of the system to handle larger numbers of users, handle larger amounts of data, and distribute the workload across multiple componentsor servers.

Maintainability:

Requirements are concerned with the ease of maintaining, modifying, and extending the software system over time. Factors to consider in software development include code maintainability, modularity, and reusability of components, quality documentation, and adherence to coding standards.

Compatibility:

Refers to the system's capability to work together and exist harmoniously with other systems, platforms, or technologies. This includes the ability to work well with certain operating systems, databases, web browsers, or the capability to communicate effectively with external systems or APIs.

3.5 TECHNOLOGY USED

3.5.1 Software Requirements

Operating System	Windows 10
Frontend	HTML, CSS, Flask Web kit Framework
Backend	Python 3.6 or higher
Database	MySQL

Table 3.5.1 Software Requirements

3.5.2 Hardware Requirements

Processor	Intel Core i3 or higher
RAM	8GB or higher
Storage	10GB or higher

Table 3.5.2 Hardware Requirements

CHAPTER 4

SYSTEM ANALYSIS

4.1 PROBLEM ANALYSIS

The problem is a result of the fact that agricultural weed populations have become more diverse over time. This variety makes it difficult for farmers to visually discriminate between cultivated plants and weeds. Due to the variety of plant species, weed identification must become increasingly complex and automated. Because it takes a lot of time and effort to visually inspect plants to locate weeds, manual examination is useless. It heavily depends on the expertise and experience of the farmers, which might differ. Human error-related ineffective weed management methods have the potential to lower crop production. Farmers need a method of weed identification that is more precise and efficient if they are to properly target and eradicate weed infestations. Weed management procedures may be sped up with the use of automated weed identification techniques, which can save manpower costs and boost overall effectiveness. The size and complexity of the weed identification training dataset might be important, especially when there is a diverse population of weeds. The collection and classification of a large dataset containing all possible weed species can be challenging and time-consuming.

STUDY OF FEASIBILITY

A vital step in assessing the viability and practicality of a proposed project or initiative is the feasibility study. To assess if it is worthwhile to move on with the project, a variety of aspects are examined, including technical, economic, legal, operational, and scheduling concerns.

Economic Feasibility

The evaluation of the project's financial viability is the main goal of economic feasibility. It entails analyzing the project's advantages and disadvantages while taking into consideration the upfront expenditures, ongoing expenses, possible income sources, and return on investment (ROI). Economic viability is frequently evaluated using financial analysis techniques including cost-benefit analysis and net present value (NPV) calculations.

Technical Feasibility

The technical viability of the proposed project is examined in this component. It determines if the infrastructure, knowledge, and technology necessary for the project's effective conception and execution are easily accessible. It is crucial to take into account how well new technology will integrate with current systems, technological hazards, and other factors.

Operational Feasibility

Operational viability assesses how quickly and effectively a project may be implemented and incorporated into current company practices. It evaluates elements such the availability of resources, the skill of the workforce, the likelihood of process interruptions, and the overall effect on daily operations. The concept must be practical for the company and easy to incorporate into ongoing operations.

CHAPTER 5

SYSTEM DESIGN

5.1 INTRODUCTION

The process of developing a software system's behavior, as well as its structure, components, and linkages, is referred to as system design and encompasses all of these facets. This process is called "work." As a component of this stage, the formulation of a comprehensive plan that details the development and implementation of the system based on the requirements that have been gathered is included. During this phase, the major attention is given on the creation of a solution that is capable of meeting the needed levels of functionality, performance, reliability, and scalability. During this phase, the development of a solution that is capable of meeting the required levels of scalability is also a primary priority.

5.2 Architecture Design

The process of developing the fundamental structure and organization of a software system is referred to as the architectural design process. This process may be broken down into several steps. It is in charge of defining the system's overarching structure, as well as its linkages, the flow of information and control within the system, and so on. The specific architecture pattern, such as client-server, layered, or micro services, has an impact not only on the way the components of the system are organized but also on their capacity to communicate with one another.

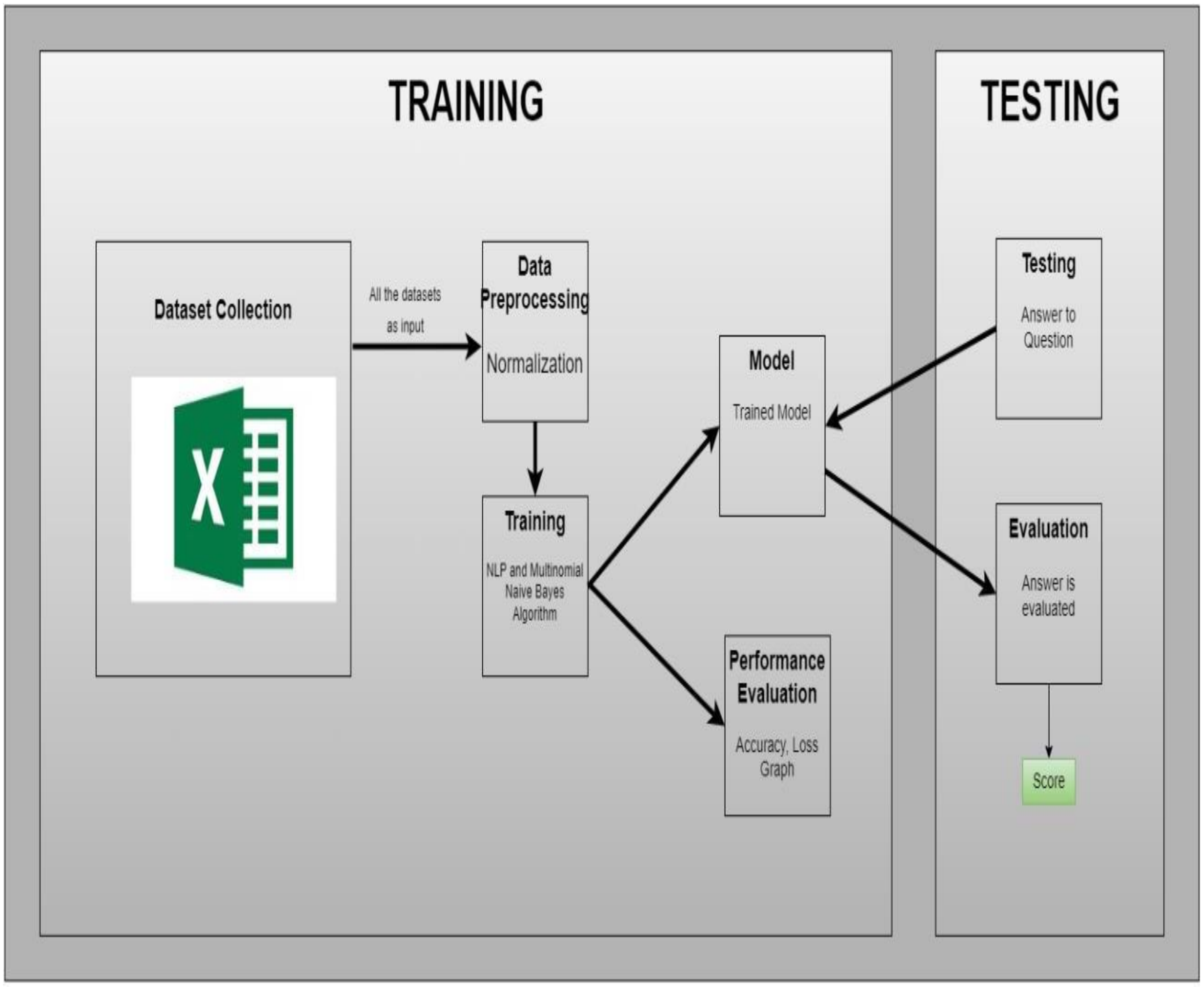


Fig 5.2: Architecture Design

Detailed Design

5.3 USE CASE DIAGRAM:

After use-case research has been finished, it is possible to build a use case diagram utilizing the Unified Modeling Language (UML). A use case diagram is sometimes referred to as a behavioral diagram. This method's major objective is to produce a visual illustration of how a system functions by highlighting the numerous individuals who are involved (also known as actors), the objectives that they want to achieve (which are referred to as use cases), and any linkages that may exist between these use cases. The primary goal of a use case diagram is to illustrate not just the responsibilities of various system actors but also the specific tasks that they are responsible for carrying out within the system.

Use case – Training Coordinator Module

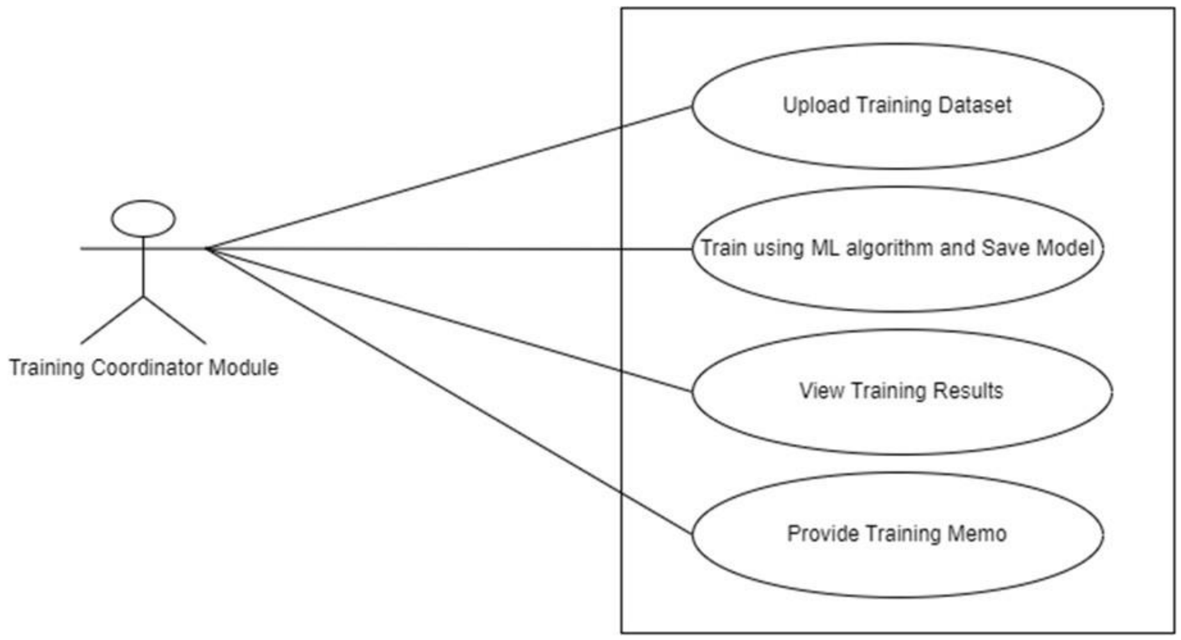

Upload Training Dataset
Train using ML algorithm and Save Model
View Training Results
Provide Training Memo
Training Coordinator Module

Fig 5.3.1 Training Coordinator Module

Use case – Evaluation Officer Module

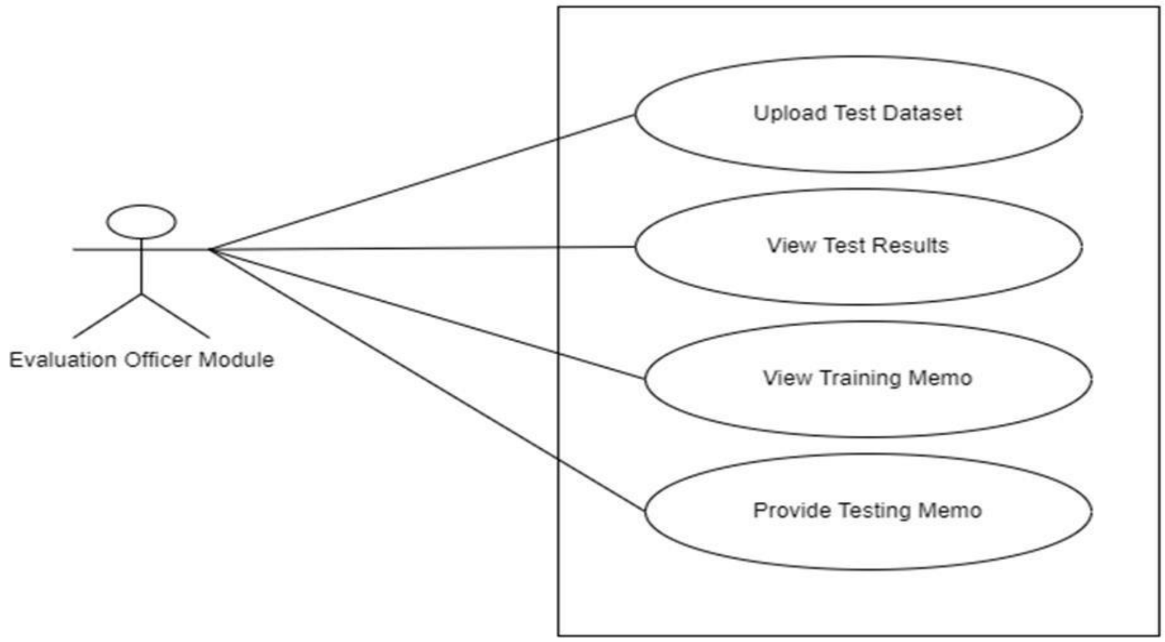


Fig 5.3.2 Evaluation Officer Module

Use case – Testing Administrator Module

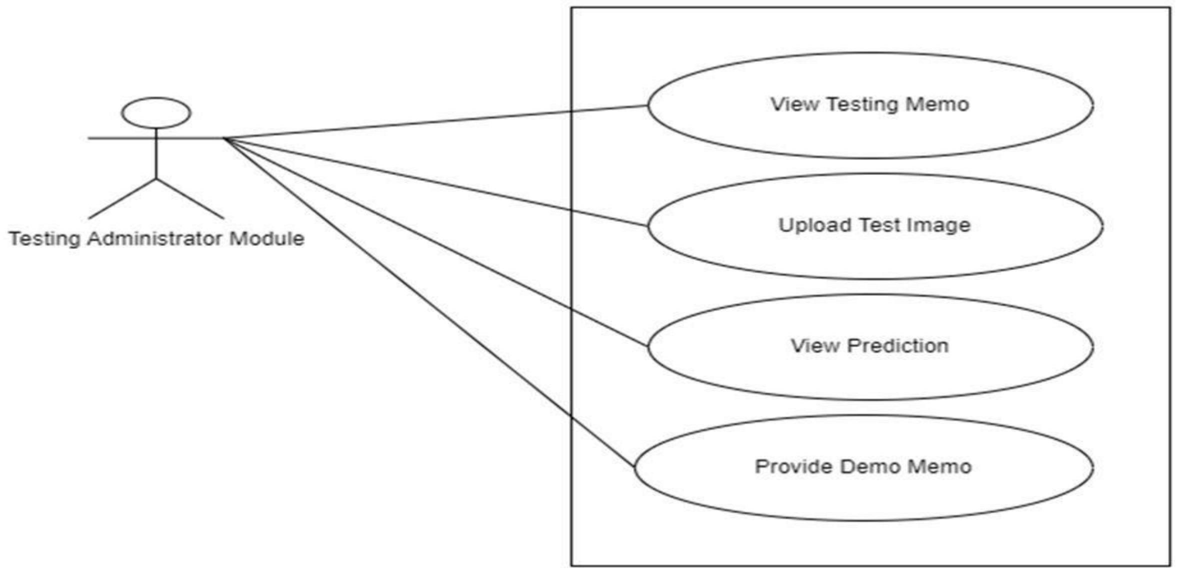


Fig 5.3.3 Testing Administrator Module

5.4 SEQUENCE DIAGRAM:

The Unified Modeling Language (UML) defines a sequence diagram as a graphical representation that shows the development and progression of operations inside a system in the order that they are performed sequentially. The preceding example exemplifies what is meant by the term "sequence diagram." A graphical representation of the many different methods in which the many actors or components communicate with one another is provided in the main text. Diagrams showing the sequence of events can also be referred to as event diagrams, event situations, or timing diagrams. Sequence diagrams are one sort of diagram. The concept being discussed here is referred to by each of these many names. They give light on the mechanism in which information or events are conveyed from one player to another inside a system, which helps us to better comprehend the chronological order in which the events that take place within that system take place.

Sequence – Training Coordinator Module

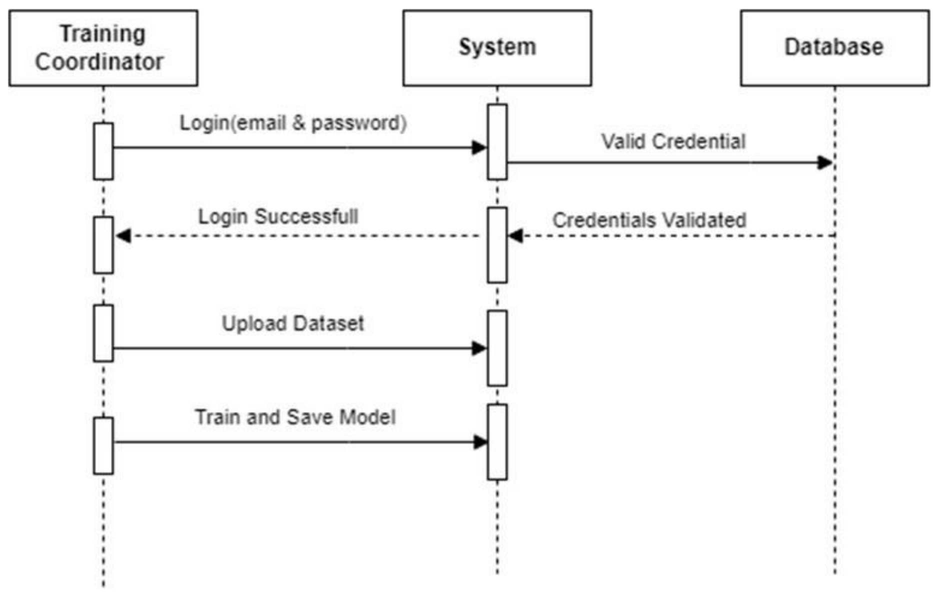


Fig 5.4.1 Sequences – Training Coordinator Module

Sequence – Evaluation Officer Module

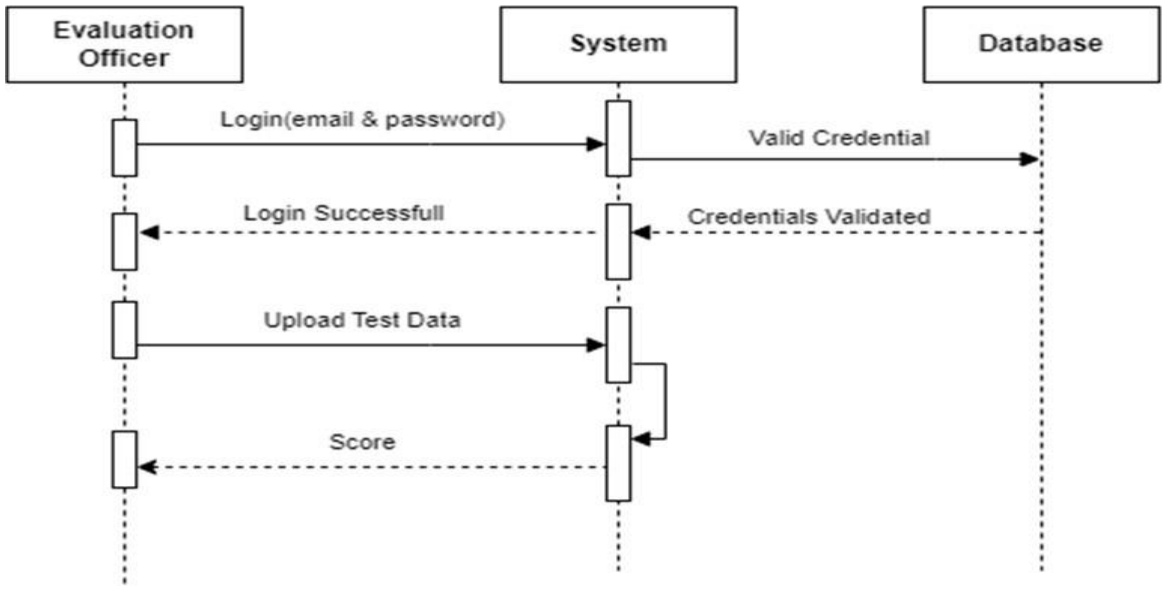


Fig 5.4.2 Sequences – Evaluation Officer Module

Sequence – Testing Administrator Module

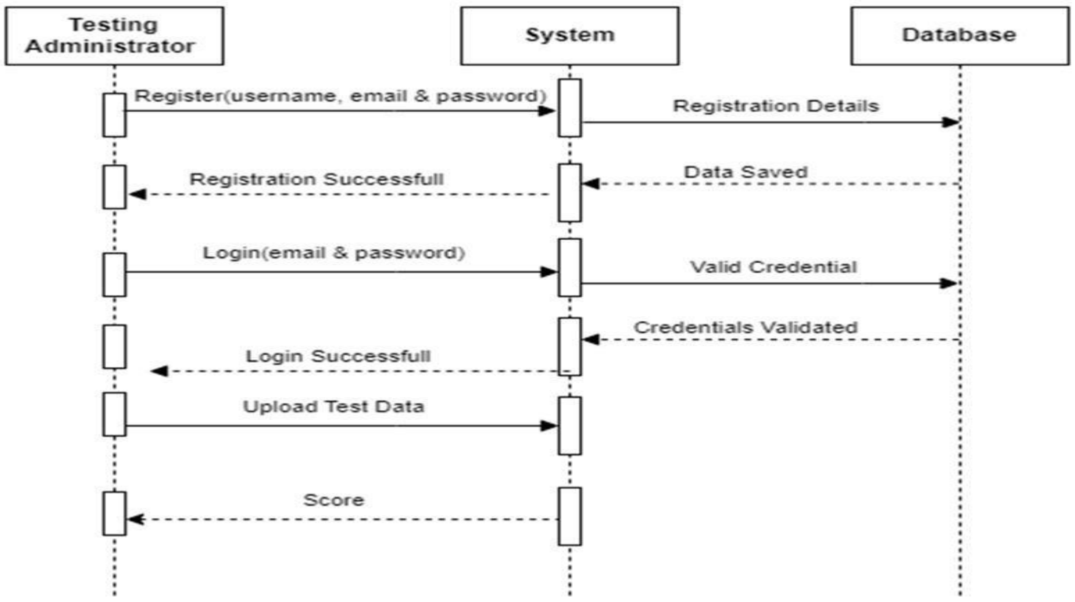


Fig 5.4.3 Sequence – Testing Administrator Module

5.5 ACTIVITY DIAGRAM:

A sort of diagram known as an activity diagram is one that is employed for the purpose of graphically depicting the activities and processes that occur in the sequence in which they occur during a process. The graphics depict not just the decision-making process but also repetition and the execution of many tasks in tandem. It is common practice to use process flow diagrams to present a well-structured and coherent portrayal of how a process evolves and the multiple activities that are a part of it. This is done in order to facilitate better understanding. Activity diagrams are a useful tool that is made available to us by the Unified Modelling Language (UML).

Activity Diagram – Training Coordinator Module

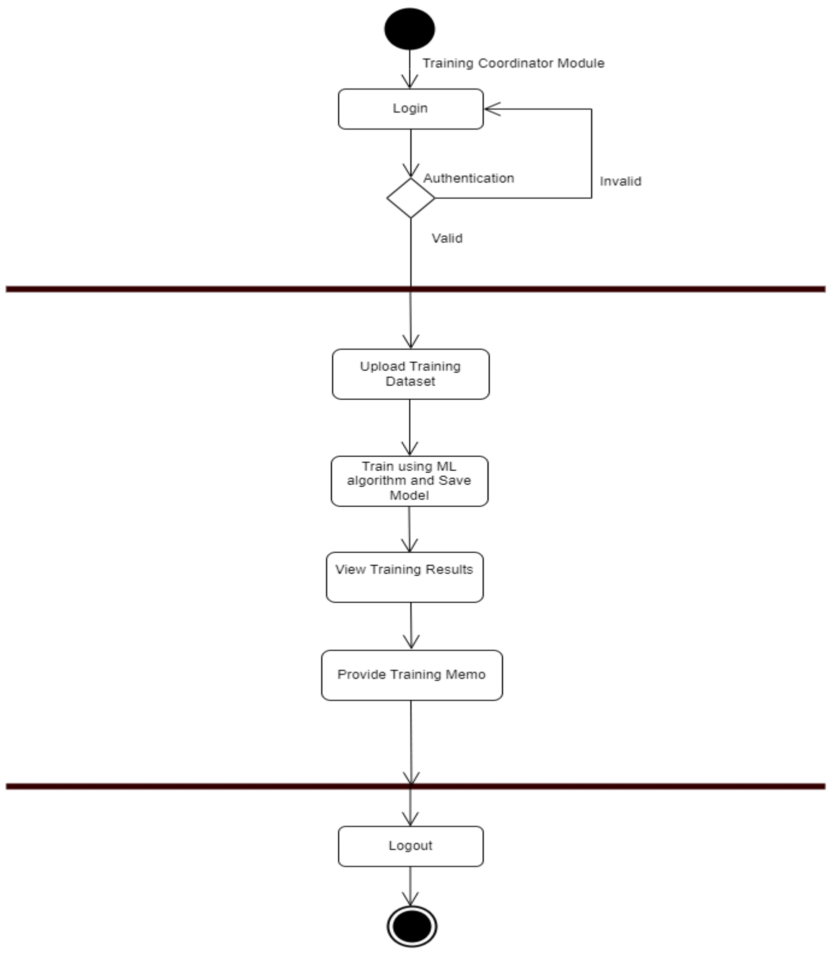


Fig 5.5.1 Training Coordinator Module

Activity Diagram – Evaluation Officer Module

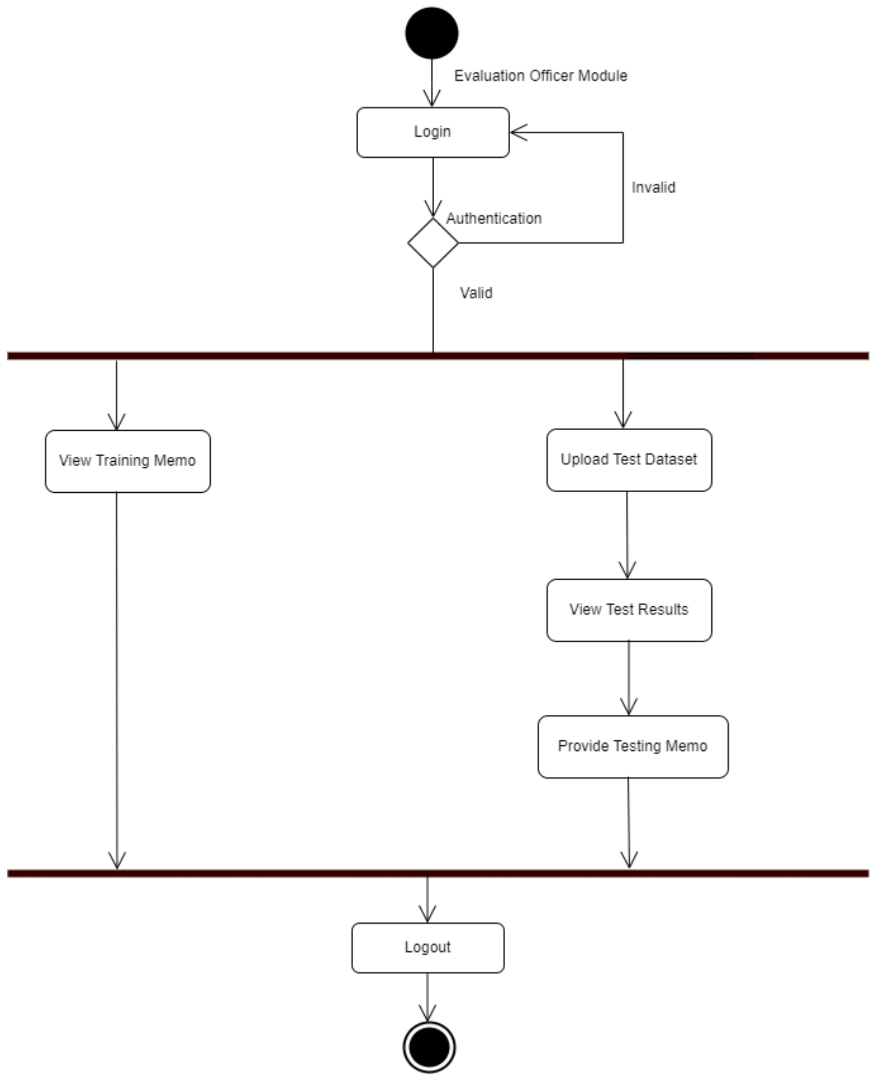


Fig 5.5.2 Evaluation Officer Module

Activity Diagram – Testing Administrator Module

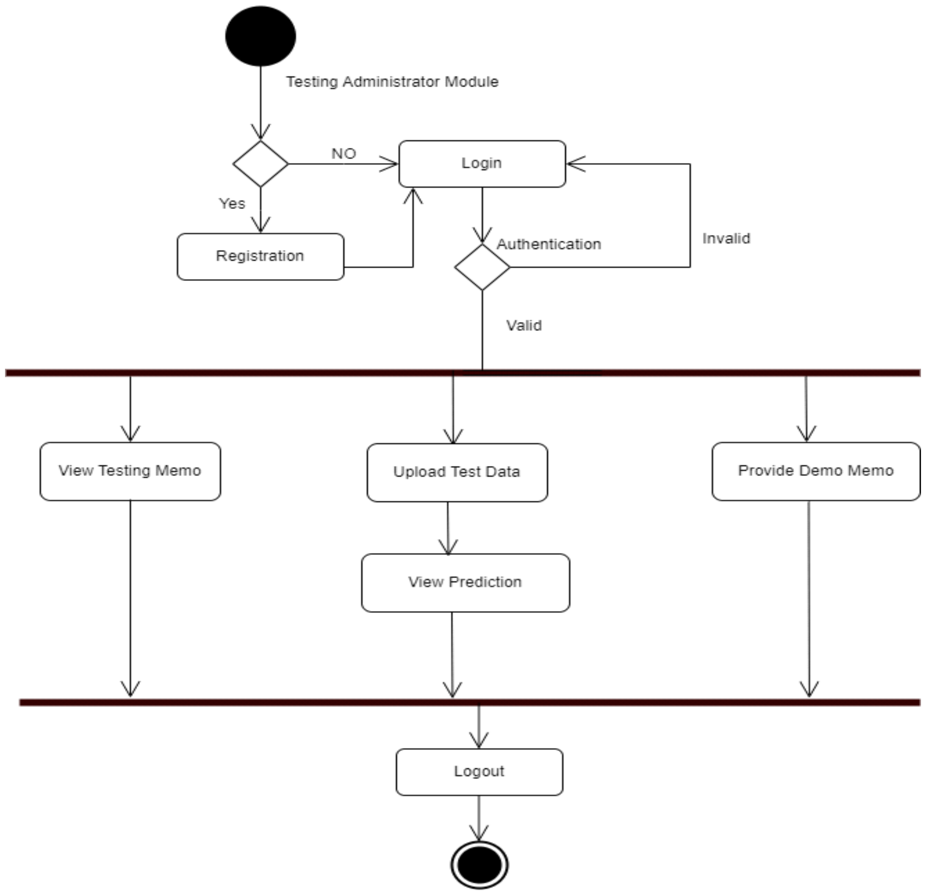


Fig 5.5.3: – Testing Administrator Module

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 INTRODUCTIONS

System implementation is the stage of the software development life cycle (SDLC) when the planned system is created, tested, and deployed. It is necessary to convert the system design into executable code before the software system may be used. The following is a list of the key steps in the system implementation process.

6.2 INPUT DESIGN

Input design refers to the process of defining how data or information is captured and entered into a software system. It involves designing user interfaces, forms, and methods that allow users to input data accurately, efficiently, and conveniently. A well-designed input system ensures that the system can effectively process and utilize the inputted data.

6.3 OUTPUT DESIGN

The process of showing information or outcomes produced by a software system to users or other stakeholders is known as output design. Designing an understandable, useful,and aesthetically beautiful structure, arrangement, and distribution plan for the system's output is necessary. Enhancing user happiness, facilitating decision-making, and effectively communicating information are the objectives of output design.

6.4 Methodology:

The proposed system's methodology involves three steps: data pre-processing, feature extraction, and classification.

Data preprocessing: The subjective answers are preprocessed to remove noise and irrelevant information. The preprocessing involves converting the text into lowercase, removing stop words, punctuation, and numbers.

Feature extraction: Relevant features are extracted from the preprocessed data. The features are based on the evaluation criteria and include sentiment analysis, semantic analysis, and lexical features.

Classification: The Multinomial Naive Bayes (MNB) model is trained on the extracted features to classify the subjective answers into different categories. The MNB model is a probabilistic model that uses Bayes' theorem to calculate the probability of each category given the features.

The performance of the proposed system is evaluated using various metrics such as accuracy, precision, recall, and F1 score. The proposed system's performance is compared with the existing evaluation methods.

CHAPTER 7

SYSTEM TESTING

7.1 INTRODUCTION

The software development life cycle (SDLC) consists of several steps, one of which is termed system testing, which plays an incredibly significant role. An in-depth investigation of a software system's whole range of operations, as well as its performance and overall level of quality, is the major objective of this task. The next step in the procedure is to perform tests on the individual components of the integrated system. This step is absolutely necessary. These tests are crucial to verifying that the system fulfils the criteria that were initially outlined and that all of its components are functioning as expected in order to get the desired results from the system.

7.2 SOFTWARE TESTING STRATEGIES

Software testing strategies are methods that are used in a systematic way to make sure that a software system meets its quality goals, follows the requirements, and works as it should. Below are a few frequently employed software testing strategies:

Functional Testing:

Is a testing technique that focuses on validating that the software system satisfies the functional requirements that are provided in the system documentation. It entails testing individual functions, features, and modules to guarantee that they perform appropriately and provide the desired outputs for the inputs that are provided. In functional testing, some of the techniques that are utilized include equivalence partitioning, boundary value analysis, and the building of test cases based on the requirements.

Integration Testing:

Integration testing is a type of software testing that checks the interactions and interfaces between various parts or modules of a software application. It tests the integrated system to make ensuring that all of its components operate together as they were designed to, that data is sent appropriately, and that interfaces are operating as expected. Approaches such as top-down testing, bottom-up testing, and sandwich testing are examples of techniques used in integration testing.

System Testing:

System testing is a procedure that evaluates the complete software system to ensure that it satisfies both functional and non-functional criteria. This evaluation may be done both manually and automatically. The objective of these tests is to analyses the entirety of the system, including all of its components and subsystems, to ensure that all of its parts are compatible with one another and do not cause any problems while working together. The process includes running tests on a variety of use cases, scenarios, and user interactions in order to validate the system's general behavior.

Acceptance Testing:

Acceptance testing is carried out so that it can be decided whether or not a software system satisfies the acceptance criteria that have been established in advance and whether or not it is appropriate for delivery to end-users or customers. Confirming the usability, functionality, and conformity of the system with the user's expectations requires testing the system using data and scenarios that reflect real-world conditions. User acceptance testing, often known as UAT, is typically carried out by end-users or other stakeholders prior to providing their final approval for system deployment.

Testing for Performance:

Testing for performance evaluates the responsiveness, scalability, stability, and resource utilization of a software system under a variety of workload situations. Performance testing is an important part of software quality assurance. It assists in locating performance bottlenecks, measuring reaction times inside the system, and determining whether or not the system satisfies performance criteria. Load testing, stress testing, and scalability testing are all examples of testing methods that are utilized in performance testing.

Testing for Security:

Testing for security focuses on locating vulnerabilities and weak points in the protection systems of a software application or system. Its purpose is to shield the system from any potential dangers while still maintaining the data's confidentiality, integrity, and accessibility. Penetration testing, vulnerability scanning, and security code reviews are some of the approaches that are utilized during security testing.

Testing for Usability:

Usability testing reviews the user interface of the software system to determine how simple it is to use, how easily it can be learned, and how satisfied users are with it. It entails keeping an eye on representative end-users while they carry out tasks and engage with the system and soliciting feedback from them at the same time. Testing for usability helps uncover usability issues, improve user experience, and enhance the usability of the system as a whole.

Testing for Regression:

When you run regression testing, you make sure that any changes or repairs you make to the software system do not result in any new flaws or regressions in the functionality that has already been tested. After modifications have been performed, it is necessary to retest certain test cases in order to validate that the computer system continues to operate as expected. During both the development and maintenance phases of the system, regression testing is helpful in maintaining the system's stability and dependability.

7.3 TEST CASES

TEST CASE NUMBER	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1.	Training Coordinator logs in to the system using a valid email and password combination for authentication and access purposes.	Login is successful	Login is successful	Pass
2.	The Training Coordinator chooses the dataset and proceeds to upload it into the system.	Dataset is uploaded successfully and is available for training	Dataset is uploaded successfully and is available for training	Pass
3.	The Training Coordinator performs the action of saving the trained model for future use.	Model is saved successfully	Model is saved successfully	Pass
4.	Training Coordinator provides training memo and saves it	Training Memo is saved successfully	Training Memo is saved successfully	Pass
5.	Evaluation Officer provides valid email and password for login purpose	Login is successful	Login is successful	Pass
6.	The Evaluation Officer uploads	Output and the results are	Output and the results are	Pass

	dataset and verifies the corresponding output for evaluation and analysis.	displayed successfully	displayed successfully	
7.	The Testing Administrator registers an account by entering their email, username, and password details.	Account is created successfully and the data is stored in the database server	Account is created successfully and the data is stored in the database server	Pass
8.	Testing Administrator logs in by providing valid email and password	Testing Administrator can login successfully	Data Analyst can login successfully	Pass
9.	Testing Administrator uploads a test dataset for testing	Score is displayed successfully	Score is displayed successfully	Pass

Table: 7.3 Test Case Scenarios

CHAPTER 8

8.1 CONCLUSION

In this research, we suggested a technique for assessing subjective responses that is based on machine learning and natural language processing (NLP). The suggested method classifies the ambiguous responses using a Multinomial Naive Bayes (MNB) model. the system's methods for feature extraction, classification, and data preparation. The suggested approach tries to increase the efficacy and validity of evaluating subjective answers. The findings demonstrate that, in terms of accuracy, precision, recall, and F1 score, the suggested technique performs better than the current assessment approaches.

8.2 FUTURE ENHANCEMENT

Future research may focus on merging multimodal data to evaluate subjective reactions. Textual data may be combined with other modalities, including pictures, charts, or videos, to accomplish this. Accurate and thorough evaluations can be made possible by multimodal evaluation, which might offer a more comprehensive knowledge of the solution.

BIBLIOGRAPHY

Journals:

[1] X. Hu and H. Xia, "Automated assessment system for subjective questions based on lsi," in 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, pp. 250–254, IEEE, 2010.

[2] M. S. M. Patil and M. S. Patil, "Evaluating student descriptive answers using natural language processing," International Journal of Engineering Research Technology (IJERT), vol. 3, no. 3, pp. 1716–1718, 2014.

[3] P. Patil, S. Patil, V. Miniyar, and A. Bandal, "Subjective answer evaluation using machine learning," International Journal of Pure and Applied Mathematics, vol. 118, no. 24, pp. 1–13, 2018.

[4] G. Jain and D. K. Lobiyal, „„„Conceptual graphs-based approach for subjective answers evaluation, """ Int. J. Conceptual Struct. Smart Appl., vol. 5, no. 2, pp. 1–21, Jul. 2017.

Text Books:

[1] "Software Engineering", Ian Sommerville,10th Edition, Pearson,2016.

[2] "Object Oriented Modelling and Design", Michael Blaha,2nd edition, Pearson,2007.

[3] "Software Testing: Principles and Practices", Srinivasan,1st Edition, Pearson,2005.

[4] "Object-oriented Software Engineering: using Java", Brugger Dutiable H, Pearson; Prentice Hall,2010.

[5] "Machine Learning, Practitioner's approach ", Josh Patterson and Adam Gibson,1st Edition, Shroff Publisher and Distributers PVT Ltd,2017

SNAPSHOTS

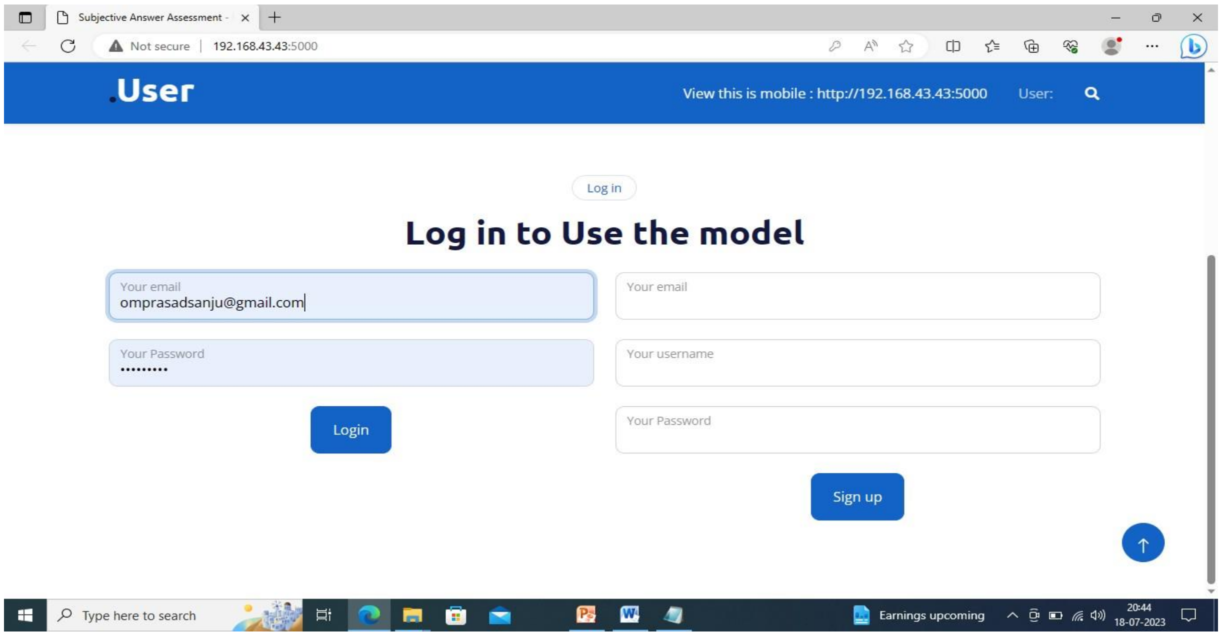
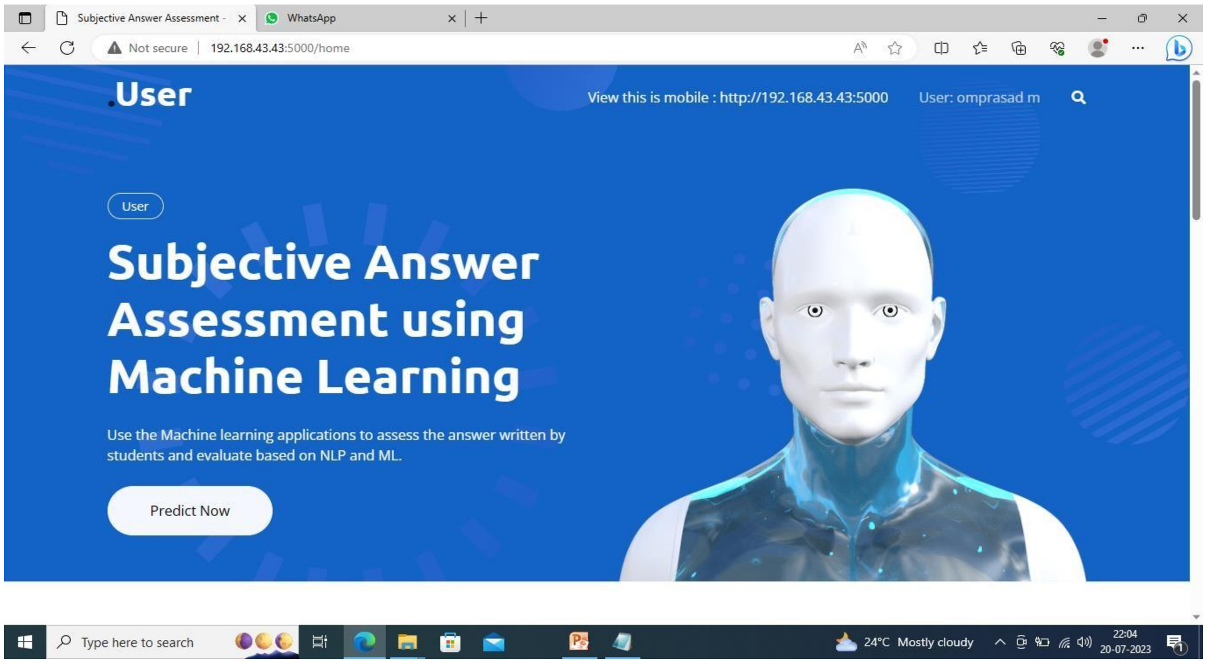
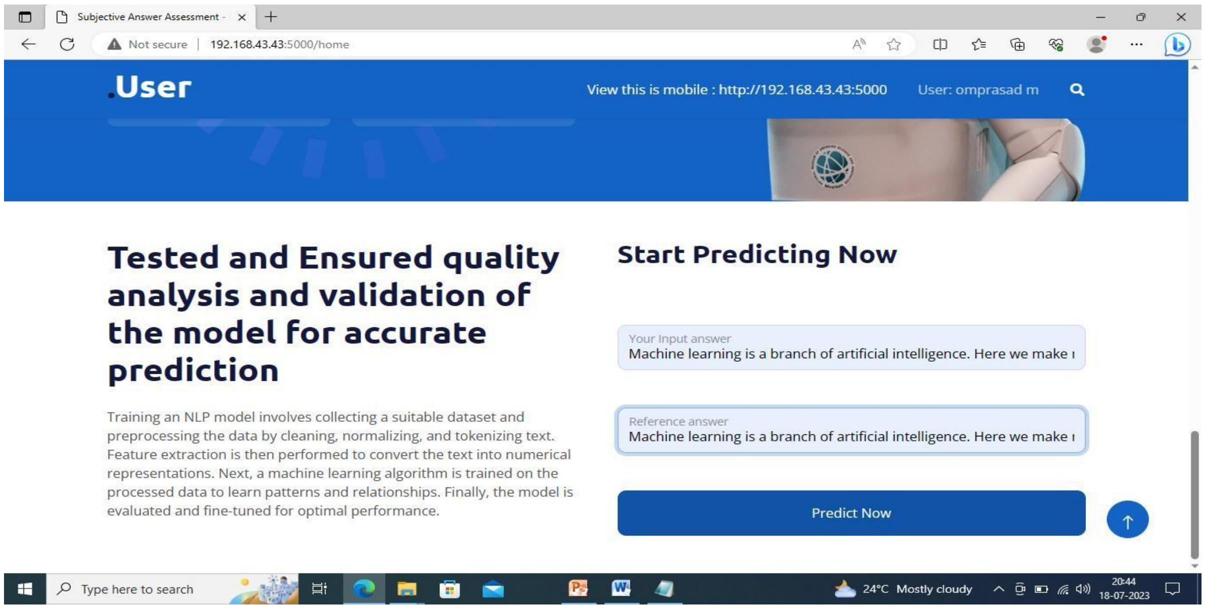


Fig (1): Login page


Subjective Answer Assessment -
WhatsApp
Not secure | 192.168.43.43:5000/home
.User
View this is mobile : http://192.168.43.43:5000
User: omprasad m
User
Subjective Answer Assessment using Machine Learning
Use the Machine learning applications to assess the answer written by students and evaluate based on NLP and ML.
Predict Now
Type here to search
24°C Mostly cloudy
22:04
20-07-2023

Fig(2): Predict Home page


Subjective Answer Assessment
Not secure | 192.168.43.43:5000/home
.User
View this is mobile : http://192.168.43.43:5000
User: omprasad m
Start Predicting Now
Predict Now
Type here to search
24°C Mostly cloudy
20:44
18-07-2023

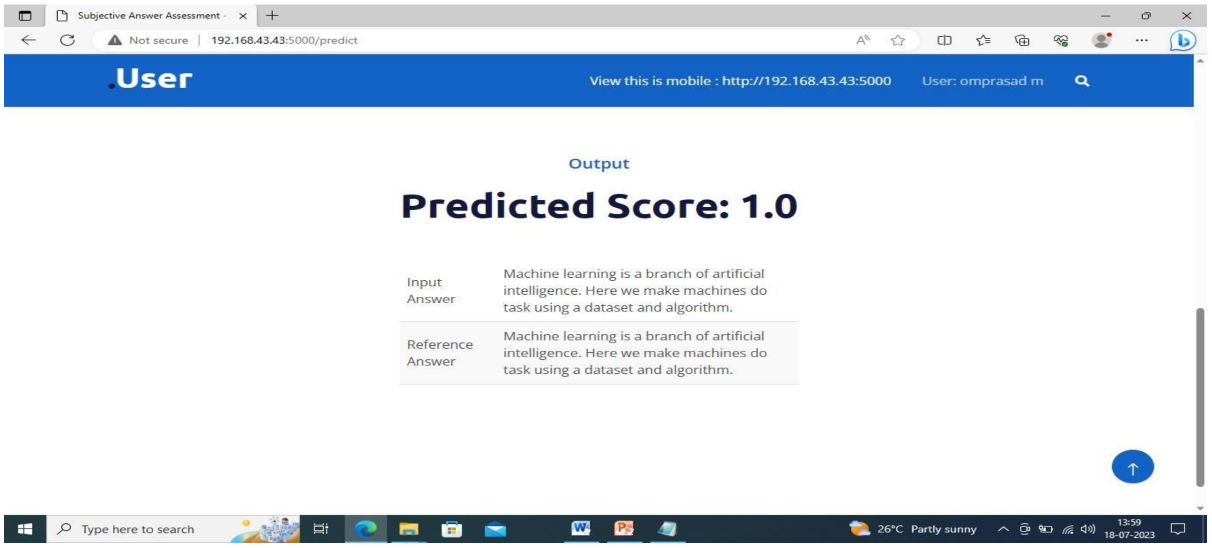
Tested and Ensured quality analysis and validation of the model for accurate prediction

Training an NLP model involves collecting a suitable dataset and preprocessing the data by cleaning, normalizing, and tokenizing text. Feature extraction is then performed to convert the text into numerical representations. Next, a machine learning algorithm is trained on the processed data to learn patterns and relationships. Finally, the model is evaluated and fine-tuned for optimal performance.

Your Input answer: Machine learning is a branch of artificial intelligence. Here we make ı

Reference answer: Machine learning is a branch of artificial intelligence. Here we make ı

Fig(3): Predicted page


Subjective Answer Assessment
Not secure | 192.168.43.43:5000/predict
.User
View this is mobile : http://192.168.43.43:5000
User: omprasad m
Type here to search
26°C Partly sunny
13:59
18-07-2023

Output

Predicted Score: 1.0

Input Answer	Machine learning is a branch of artificial intelligence. Here we make machines do task using a dataset and algorithm.
Reference Answer	Machine learning is a branch of artificial intelligence. Here we make machines do task using a dataset and algorithm.

Fig(4): Out put page