AI BASED AUTOMATIC EXAMINATION PAPER EVALUATION SYSTEM

A Project Report
Submitted in the partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by

T. RAMYA SAI	**228B1A0526**
S. HARIKADEVI	**228B1A0519**
A. M. L. TEJA SRI	**228B1A0502**
P. VIJAYA SRI	**228B1A0517**

Under the Esteemed Guidance of

Mr. G. Gopi, M. Tech

Assistant Professor

COMPUTER SCIENCE & ENGINEERING

RISE KRISHNA SAI GANDHI GROUP OF INSTITUTIONS

(Affiliated to JNTU, Kakinada)

Valluru – 523272

2022 - 2026

RISE KRISHNA SAI GANDHI GROUP OF INSTITUTIONS

(Affiliated to JNTU, Kakinada)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **"AI BASED AUTOMATIC EXAMINATION PAPER EVALUATION SYSTEM"** is a bonafide record of the work carried out by **T. Ramya Sai (228B1A0526), S. Harika Devi (228B1A0519), A. M. L. Teja Sri (228B1A0502), and P. Vijaya Sri (228B1A0517)** in partial fulfillment of the requirements for **IV Year – II Semester of Bachelor of Technology in Computer Science & Engineering, JNTUK**, during the academic year **2025–26**, under our esteemed guidance and supervision.

PROJECT GUIDE **HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the work presented in this dissertation entitled "**AI BASED AUTOMATIC EXAMINATION PAPER EVALUATION SYSTEM**" is submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering.** This project work has been carried out under the supervision of **Mr. G. Gopi,** Assistant Professor, **Rise Krishna Sai Gandhi Group of Institutions, Valluru.**

We further declare that the results embodied in this dissertation have not been submitted by us for the award of any other degree or diploma. Furthermore, the technical details furnished in the various chapters of this report are purely related to the above project, and there is no deviation from the theoretical principles followed in the design, development, and implementation.

T. RAMYA SAI	**228B1A0526**
S. HARIKADEVI	**228B1A0519**
A. M. L. TEJA SRI	**228B1A0502**
P. VIJAYA SRI	**228B1A0517**

ACKNOWLEDGEMENT

We are hereby acknowledging our profound gratitude to our beloved chairman, **Sri I. C. RANGAMANNAR GARU,** who took keen interest and encourage us is every effort throughout the course.

We express our special thanks to our secretary and correspondent of RISE Krishna Sai Gandhi Group of Institutions, **Sri SIDDA HANUMANTHA RAO GARU,** for making our project complete.

We owe our gratitude to our Principal, **Dr. K. V. SUBRAHMANYAM GARU,** for his kind attention in the valuable guidance to us throughout the course.

Words are poor gratitude to bearers but I take this opportunity to offer other sincere gratitude to our esteemed Head of the Department, **Mr. CH. HARI KRISHNA GARU** for his constant valuable suggestions help and noble guidance. Without this project would have never seen light of success.

We sincerely thank our guide **Mr. G. GOPI,** Assistant professor, for her guidance and valuable suggestions and support in the completion of the project. Finally, we thank all faculty members, supporting staff of CSE department and friends for their king cooperation and valuable help for completing the project.

ABSTRACT

Evaluating subjective papers manually is a difficult and exhausting process. One of the main problems in using Artificial Intelligence (AI) to analyze these papers is the lack of proper understanding and acceptance of the data involved. There have been several efforts to automatically score students' answers using computer science methods. However, most of these approaches rely on traditional word counts or specific keywords to accomplish this. Additionally, there is a shortage of well-curated data sets for this purpose. This paper introduces a new method that makes use of various machine learning, natural language processing techniques, and tools like Wordnet, Word2vec, word mover's distance (WMD), cosine similarity, multinomial naïve bayes (MNB), and term frequency-inverse document frequency (TF-IDF) to automatically assess descriptive answers. The evaluation process involves analyzing solution statements and key terms, and a machine learning model is trained to predict the scores assigned to the answers. The results indicate that WMD performs better than cosine similarity in general. With sufficient training, the machine learning model can also function as a standalone tool. The experiment achieved an accuracy of 88% without using the MNB model. The error rate was further reduced by 1.3% when the MNB model was included.

CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Subjective questions and answers allow for a more comprehensive assessment of a student's performance and understanding in an open-ended format. Unlike objective questions, these answers are not restricted by any particular constraints, giving students the freedom to express their thoughts and ideas based on their own interpretations and knowledge of the subject. However, there are several key differences that distinguish subjective answers from their objective counterparts. Firstly, they tend to be longer in length. Secondly, they require more time to write. Additionally, they involve a greater amount of context and demand a higher level of concentration and objectivity from the evaluator.

Assessing these types of questions using computers is a complex and challenging task, primarily due to the inherent ambiguity of natural language. To make the process manageable, several preprocessing steps are necessary, including data cleaning and tokenization before any analysis can be conducted. The textual data can then be compared using various methods such as document similarity, latent semantic structures, concept graphs, and ontologies. The final score is typically determined based on factors like similarity, presence of keywords, structure, and language use. While many attempts have been made in the past to address this issue, there is still much room for improvement, some of which is explored in this paper. Subjective exams are often perceived as more complex and intimidating by both students and teachers because of their core characteristic—context. A subjective answer requires the evaluator to carefully examine every word to assign a score, and the evaluator's mental state, fatigue, and objectivity significantly influence the final outcome.

Therefore, it is more efficient in terms of time and resources to allow a system to handle the difficult and critical process of evaluating subjective answers. Evaluating objective answers with machines is straightforward and practical. A program can be provided with questions and one-word answers, enabling a quick matching of student responses. On the other hand, subjective answers are far more challenging to process. They vary in length and contain an extensive range of vocabulary. Moreover, students often use synonyms and abbreviations, which further complicates the evaluation process.

1.1 MOTIVATION

The motivation behind this project is to reduce the time, effort, and subjectivity involved in the manual evaluation of examination papers. Traditional paper evaluation is a tedious and error-prone process that depends heavily on human judgment, which can vary from evaluator to evaluator. With the increasing number of students and examinations, there is a strong need for an automated, accurate, and unbiased evaluation system. By using Artificial Intelligence, Machine Learning, and Natural Language Processing techniques, this project aims to make the evaluation process faster, more consistent, and

more reliable. The system is designed to help teachers and institutions focus more on teaching and learning activities rather than spending excessive time on paper correction. Ultimately, it seeks to improve transparency, efficiency, and fairness in the assessment process.

1.2 PROJECT DEFINITION

The major problem addressed by this project is the inefficiency and inconsistency in manual evaluation of subjective examination papers. Evaluating descriptive answers requires a significant amount of time and concentration, and the results may vary depending on the examiner's experience, fatigue, and personal judgment. Moreover, handling large volumes of answer scripts increases the chances of errors and delays in result processing. Therefore, there is a need for a system that can understand the context, meaning, and relevance of student answers. This project aims to solve this problem by developing an AI-based automatic examination paper evaluation system that can analyze and compare student answers with model answers using NLP and ML techniques to provide accurate and fair grading.

1.3 OBJECTIVE OF THE PROJECT

The main objective of this project is to design and develop an AI-based automatic examination paper evaluation system that can evaluate subjective answers efficiently and accurately. The system aims to use Machine Learning and Natural Language Processing techniques to understand the semantic similarity between student responses and the expected answers. It seeks to automate the grading process by reducing human effort, minimizing evaluation bias, and speeding up result generation. The project also aims to ensure consistency in assessment, provide reliable scoring, and handle large-scale evaluations effectively. Overall, the objective is to create a smart evaluation platform that enhances the quality, accuracy, and efficiency of the examination process.

2. LITERATURE SURVEY

1. **Title: "Automated Subjective Answer Evaluation Using Generative AI and Machine Learning"**

   Author: Divekar Nikhil et al., 2024

   Description: This paper presents a Java-based web application that uses Generative AI and Machine Learning for automatic checking of subjective answer sheets. It utilizes algorithms such as Keyword Matching, Sentence Matching, Levenshtein Distance, and Heuristic Rule-Based Comparison to measure similarity between student answers and model answers. The system aims to replicate human evaluation by providing reliable similarity-based scoring for long-answer questions. The study shows that the approach reduces time, effort, and bias in manual checking while maintaining consistency and accuracy in evaluation.

2. **Title: "Artificial Intelligence in Open Book Examination Systems"**

   Author: Amit Dimari et al., 2025

   Description: This paper explores the application of Artificial Intelligence in open book examination systems using Natural Language Processing (NLP) and Machine Learning techniques. It employs statistical analysis through the t-test along with AI-based evaluation models to compare traditional and automated assessment methods. The study demonstrates that AI-integrated systems significantly improve feasibility, reduce subjectivity, and enhance the efficiency of evaluation. It also highlights AI's role in dynamic question paper preparation and automated answer evaluation, showing how intelligent systems can modernize educational assessments and ensure fairness and accuracy in performance analysis.

3. **Title: "Automated Evaluation System for Exam Paper Quality Analysis Using Text Mining and NLP"**

   Author: Doaa Mohamed Elbourhamy, 2025

   Description: This paper introduces an automated evaluation system that analyzes the formal and technical quality of university exam papers using Text Mining, NLP, and Machine Learning. It applies TF-IDF for feature extraction, the Naïve Bayes classifier for classification, and rule-based text matching for performance evaluation. The system achieved an accuracy of around 98%, proving its effectiveness in improving exam paper quality.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system for subjective answer evaluation is mainly based on Machine Learning and Natural Language Processing techniques. It uses methods such as tokenization and lemmatization for text preprocessing, and text representation techniques like Bag of Words, TF-IDF, and Word2Vec to convert textual answers into numerical form. For measuring the similarity between student answers and model answers, techniques such as Cosine Similarity and Word Mover's Distance are commonly used. In addition, classification algorithms like Multinomial Naive Bayes are applied to predict or categorize the quality of answers.

To evaluate the performance of these models, standard metrics such as Accuracy, F1-score, and Recall are used. These methods have helped in automating the evaluation process and reducing manual effort, but they still have several limitations when handling subjective answers.

Major limitations of the existing system include:

- Existing systems struggle to effectively handle synonyms and variations in word usage, which can affect similarity measurement.
- Subjective answers often vary widely in length, making it difficult to maintain consistent evaluation accuracy.
- The order of sentences in answers may differ, and current approaches may not correctly capture meaning when sentences are randomly arranged.

Because of these challenges, the existing system is not always able to accurately reflect the true semantic similarity between the student's answer and the reference answer, leading to less reliable evaluation results.

3.2 PROPOSED SYSTEM

The proposed system introduces an enhanced and automated method for evaluating descriptive answers using advanced Machine Learning and Natural Language Processing techniques. It follows a two-step evaluation approach to ensure both accuracy and flexibility in assessment.

In the first step, the student's answer is compared with the model answer and the predefined keywords using similarity-based techniques such as Word Mover's Distance and semantic similarity measures. This stage focuses on understanding the contextual meaning of the answer rather than just matching exact words. It evaluates how closely the student's response aligns with the expected concept while also checking for the presence of important keywords.

In the second step, the results obtained from the similarity evaluation are used to train a machine learning model. Once trained, this model can evaluate student answers even when model answers and keywords are not explicitly provided. This makes the system more scalable and efficient for large-scale examinations.

For example, consider the question:

"What is the capital city of Pakistan and what is it famous for?"

Model Answer:

"Islamabad is the capital city of Pakistan and it is famous for mountain scenery."

Keywords:

Islamabad, mountain scenery

The system processes the question, model answer, student answer, and keywords. It assigns marks based on two factors:

- Contextual similarity between the student answer and the model answer
- Presence or absence of important keywords

Sample evaluations:

- "Karachi is the capital of Pakistan, it is famous for mountain scenery" → 50% marks (partial keyword match, incorrect context)
- "Islamabad and mountain scenery" → 30% marks (keywords present but lacks context)
- "Islamabad is the capital and it is famous for mountain scenery" → 100% marks (both context and keywords are correct)

This approach ensures fair and meaningful evaluation by considering both semantic understanding and essential concept coverage.

3.3 SOFTWARE REQUIREMENT SPECIFICATIONS

HARDWARE REQUIREMENTS

Minimum (Required for Execution)

- **System:** Pentium IV 2.2 GHz
- **Hard Disk:** 20 GB
- **RAM:** 1 GB

My System (Development Environment)

- **System:** Intel i3 Processor (5^{th} Generation)
- **Hard Disk:** 500 GB
- **RAM:** 4 GB

SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10 / Windows 11
- **Development Software:** Python 3.10
- **Programming Language:** Python
- **Domain:** Machine Learning
- **Integrated Development Environment (IDE):** Visual Studio Code
- **Front End Technologies:** HTML5, CSS3, JavaScript
- **Back End Framework:** Django
- **Database Language:** SQL
- **Database (RDBMS):** MySQL
- **Database Software:** WAMP or XAMPP Server
- **Web Server / Deployment Server:** Django Application Development Server
- **Design / Modelling Tool:** Rational Rose

3.4 FEASIBILITY STUDY

A feasibility study is conducted to evaluate whether the proposed system is practical, beneficial, and viable before proceeding to the design and implementation phases. It helps in determining if the project can be successfully developed using the available resources, technology, and budget. The feasibility study serves as a preliminary investigation that enables management to decide whether to continue with the project or consider alternative solutions.

From a system analyst's perspective, feasibility analysis is a key decision-making tool that assesses the overall suitability of the project. It ensures that the proposed system is technically possible, operationally acceptable, and economically affordable. The main objective is to confirm that the project will add value to the organization and meet user requirements effectively.

The feasibility study for this project is analyzed under three major aspects:
Technical Feasibility, Operational Feasibility, and Economic Feasibility.

1. **Technical Feasibility** – The proposed system is technically feasible as it is developed using reliable and well-established technologies such as Python, Django, and Machine Learning. All required software tools are easily available and can run on standard computer systems. The hardware requirements are minimal and can be met by existing systems. The NLP and similarity techniques used in the project are proven and suitable for automated answer evaluation. Therefore, the system can be developed without any technical difficulty.

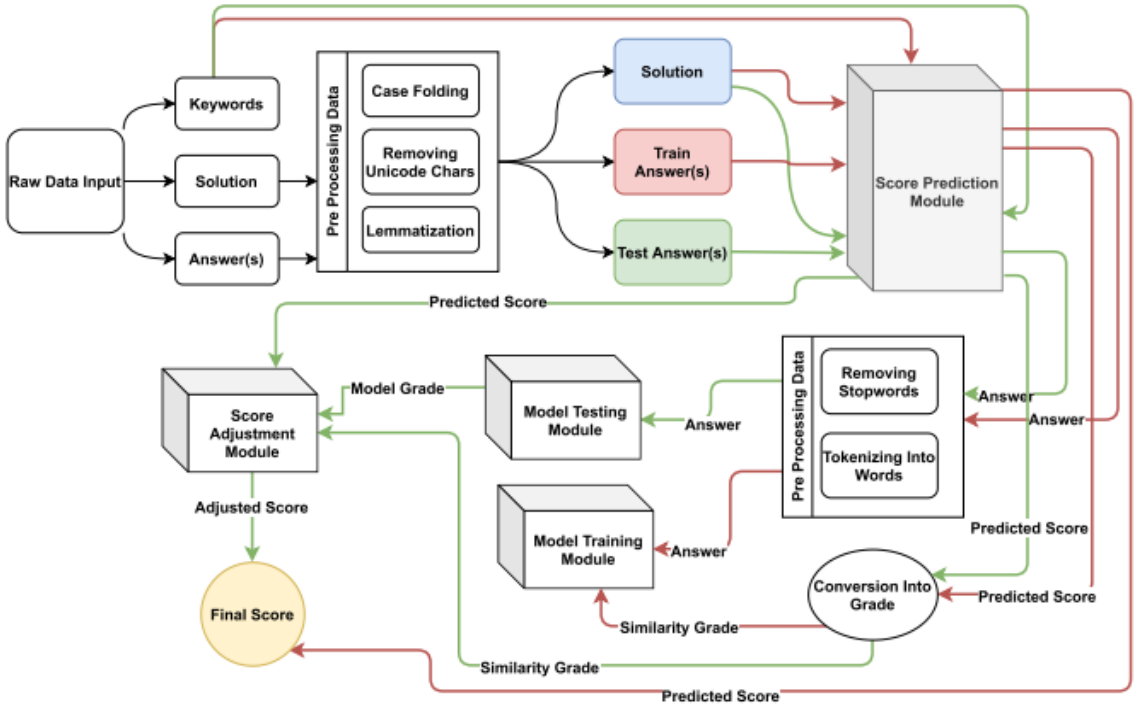
2. **Operational Feasibility** – The system is operationally feasible because it is simple to use and significantly reduces the manual effort involved in evaluating descriptive answers. It automates the evaluation process, saves time, and ensures consistency in results. The user interface is easy to understand, and no special training is required. Since the system improves efficiency and accuracy, it is likely to be accepted by both teachers and management.

3. **Economic Feasibility** – The system is economically feasible as most of the tools and technologies used are open-source and free of cost. There is no need for additional hardware investment. The development and maintenance costs are minimal compared to the benefits obtained. By reducing manpower and saving time, the system proves to be cost-effective in the long run.

By analyzing these three factors, the feasibility study ensures that the project is realistic, sustainable, and capable of achieving its intended objectives.

4. SYSTEM DESIGN

4.1SYSTEM ARCHITECTURE



4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

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

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

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

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

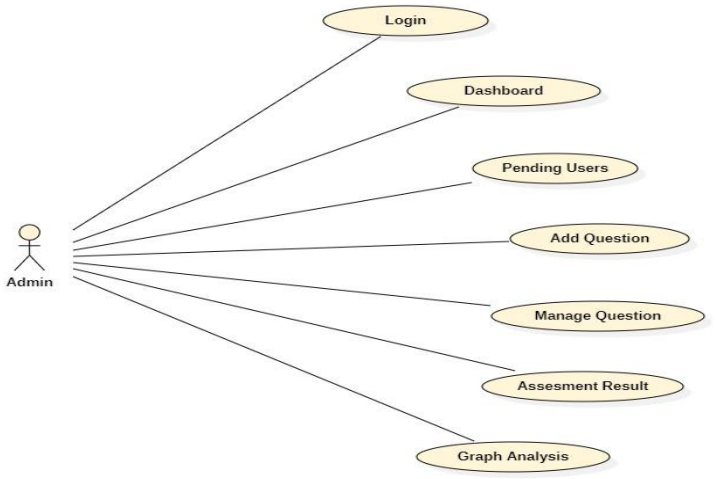
5. Encourage the growth of OO tools market.

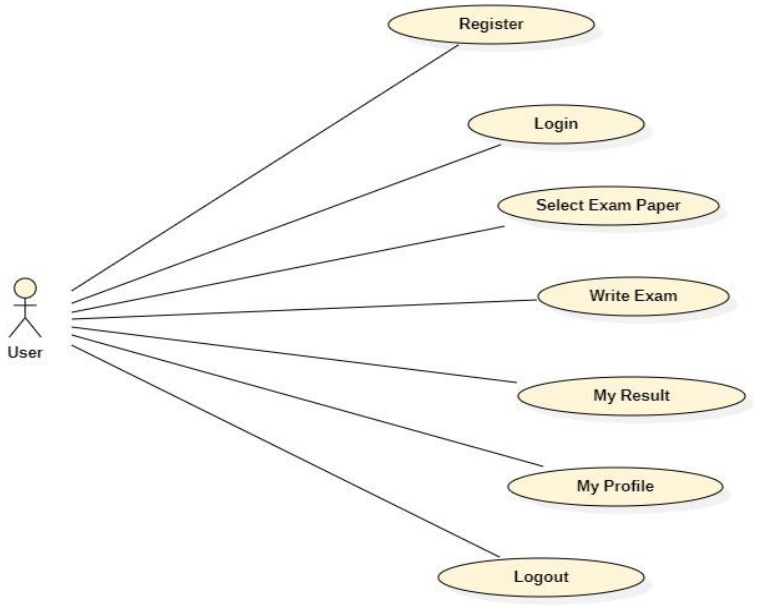
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

7. Integrate best practices.

4.2.1 USE CASE DIAGRAM

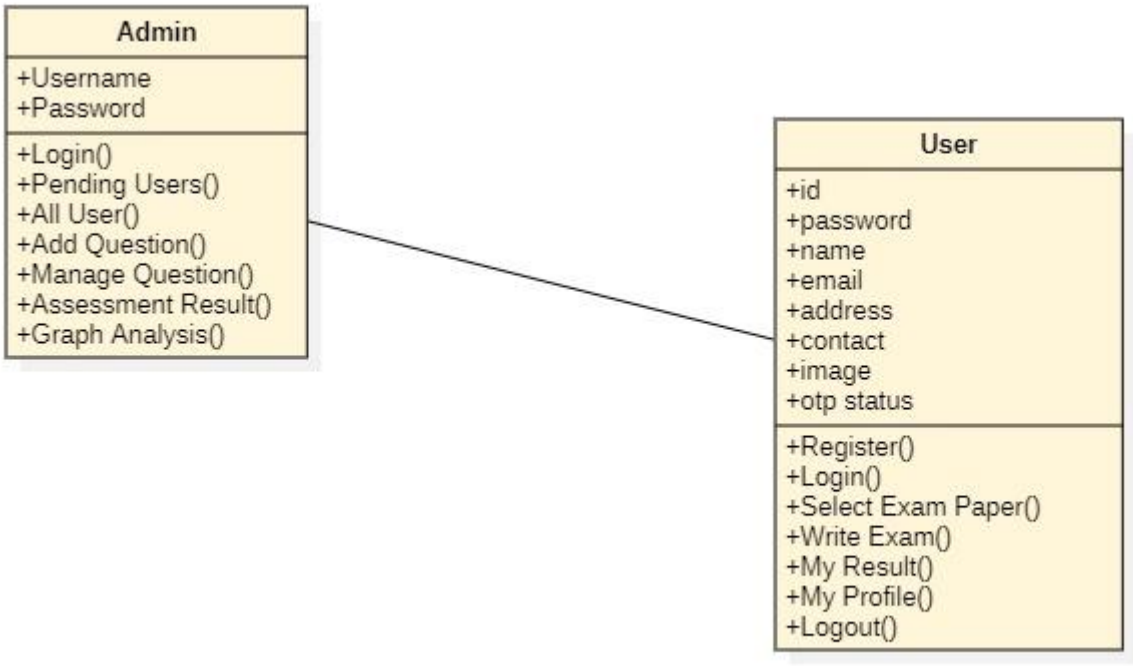
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.





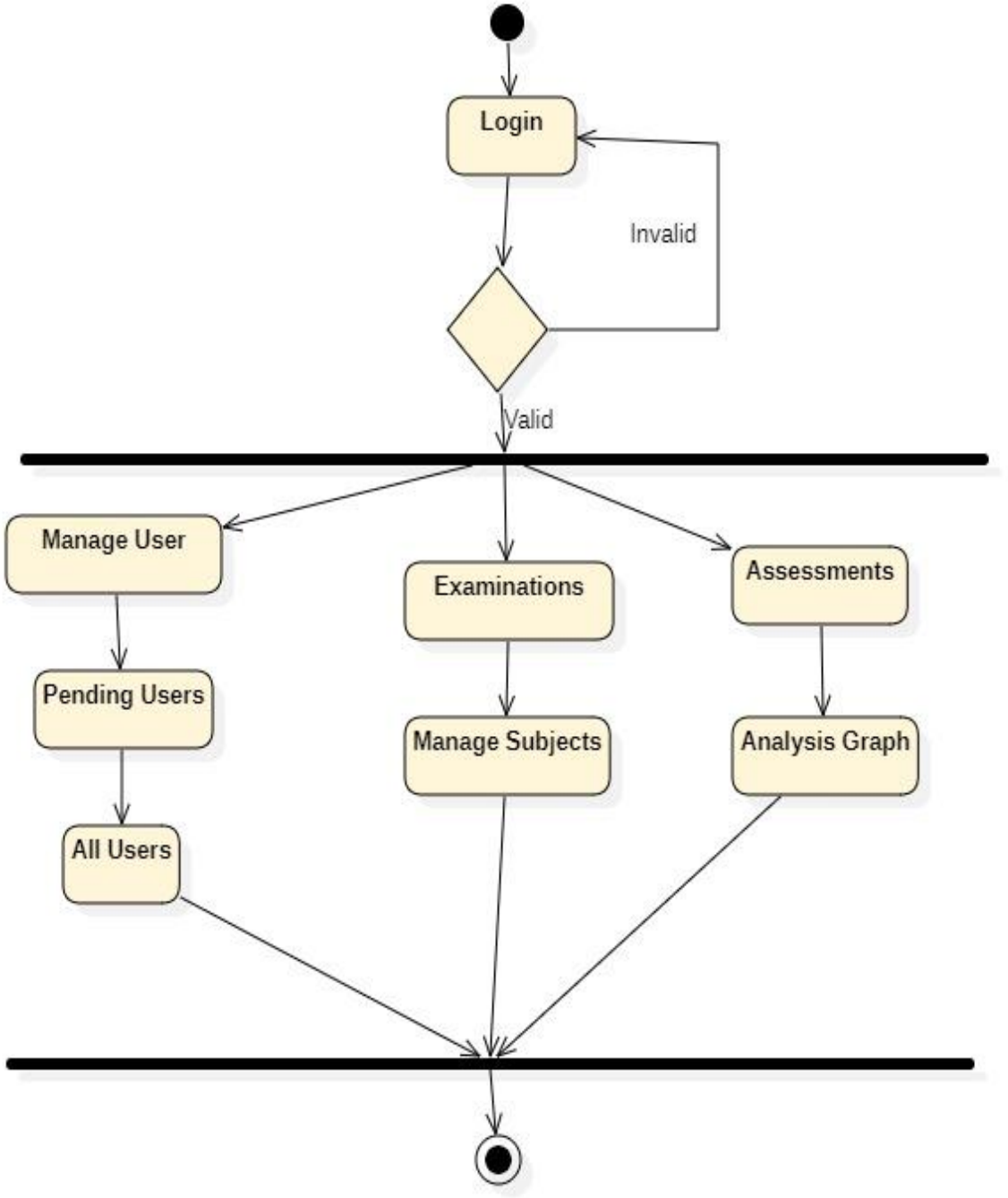
4.2.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

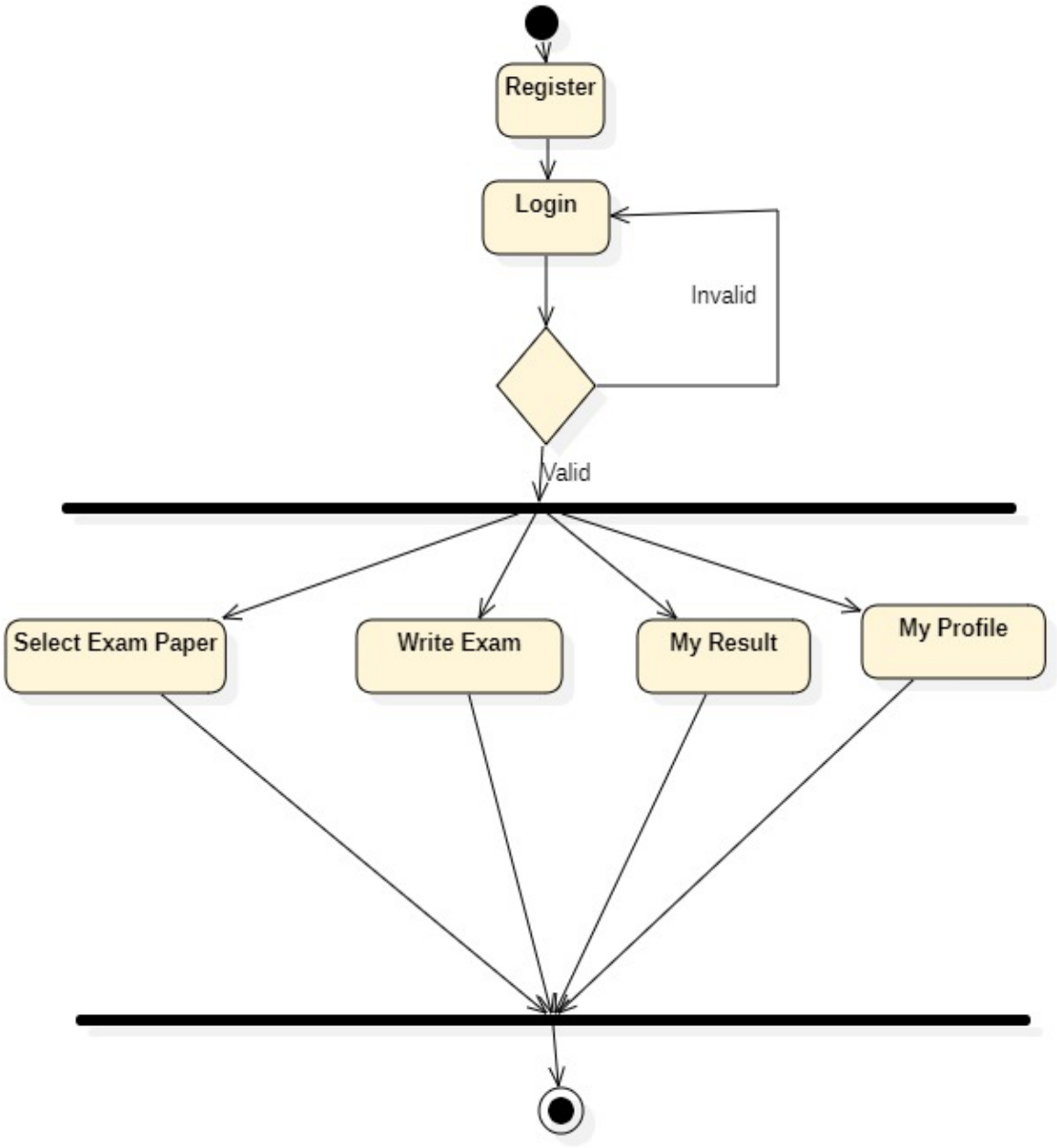


4.2.3 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



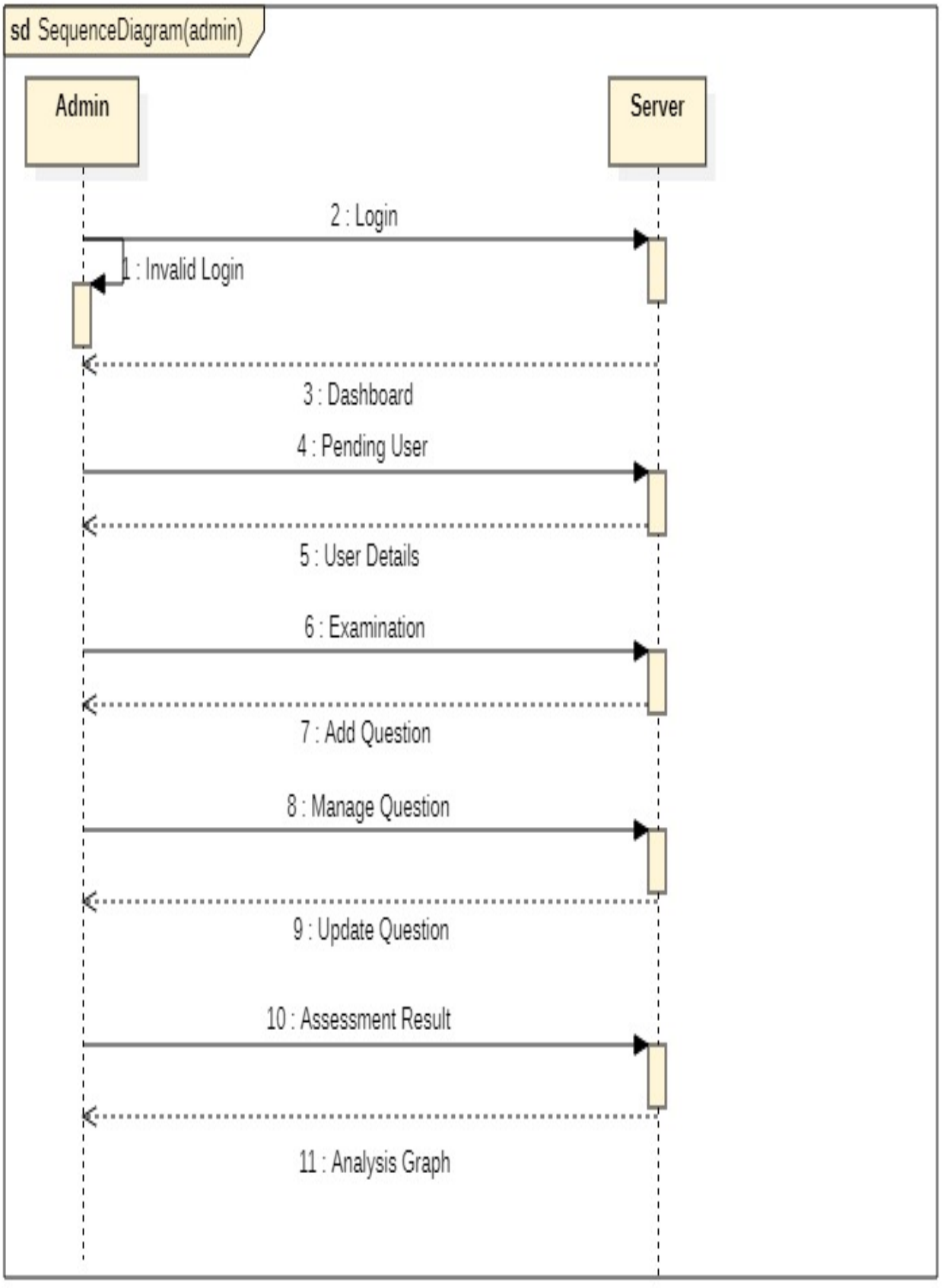
Activity Diagram (Admin)



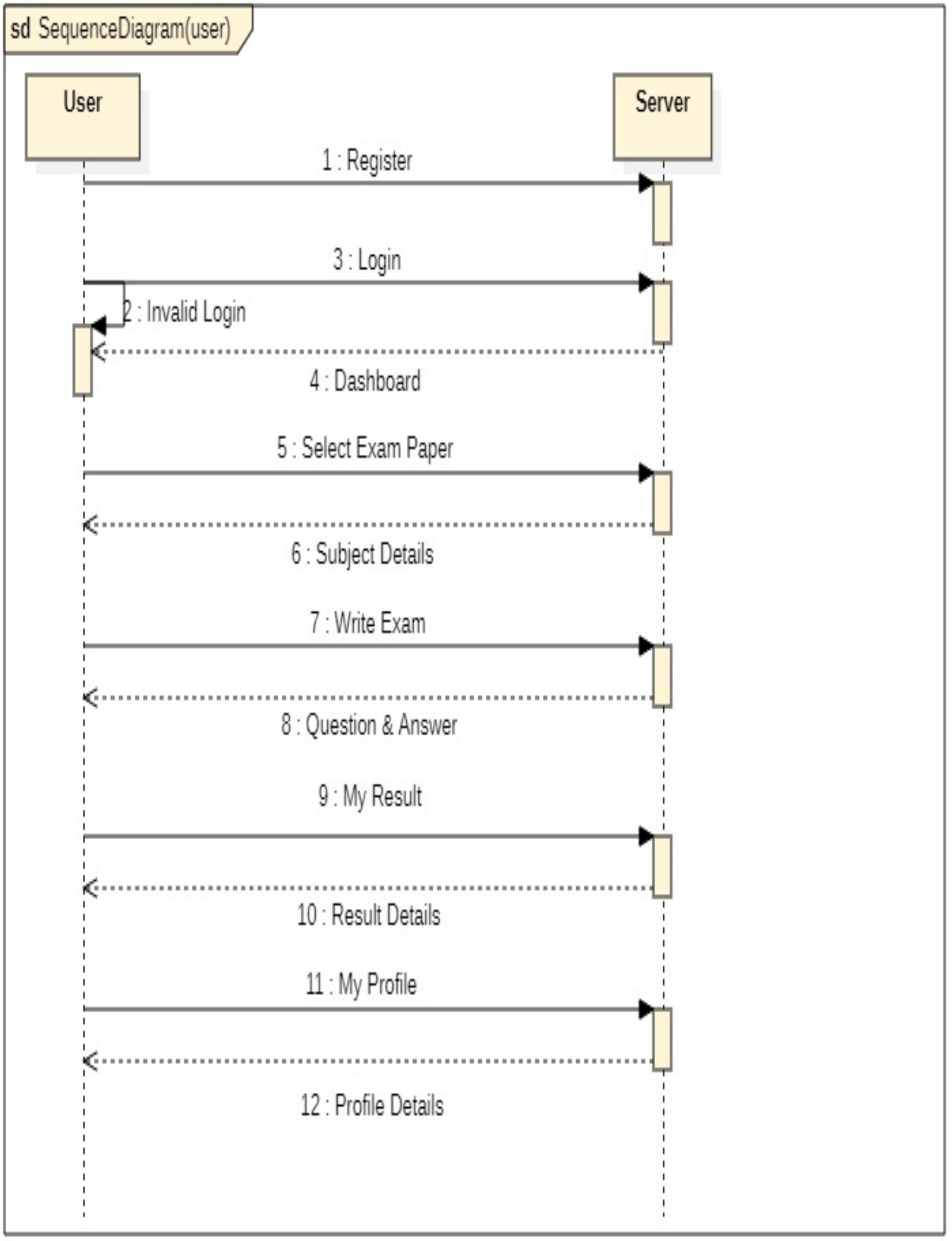
Activity Diagram (User)

4.2.4 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



Sequence Diagram (Admin)



Sequence Diagram (User)

5. IMPLEMENTATION

5.1 ABOUT PYTHON

Python is a high-level, interpreted, and general-purpose programming language that is widely used in web development, machine learning, artificial intelligence, data science, and automation. It supports both object-oriented and procedural programming paradigms. Python is known for its simple syntax, high readability, and fast development process. Programs written in Python are shorter and easier to understand when compared to many other programming languages such as Java and C++.

Python is chosen for this project because it provides excellent support for Machine Learning and Natural Language Processing. It has a rich set of libraries such as NumPy, Pandas, Scikit-learn, NLTK, TensorFlow, and Keras which simplify complex operations like data preprocessing, model training, and similarity computation. Integration with Django makes Python ideal for developing web-based applications, allowing easy connection between the frontend and backend.

Python increases developer productivity because fewer lines of code are required to implement functionalities. Debugging and maintenance are easier due to its clean syntax and dynamic typing. It is also platform-independent, meaning the same code can run on Windows, Linux, and macOS without modification.

Advantages of Python:

- Simple and easy to learn syntax
- Large collection of libraries and frameworks
- Faster development and reduced coding effort
- Strong support for Machine Learning and AI
- Open-source and free to use
- Portable across different platforms

Disadvantages of Python:

- Slower execution speed compared to compiled languages
- Not widely used for mobile application development
- Less suitable for applications requiring very high performance

Despite these limitations, Python remains the best choice for this project because its advantages strongly support intelligent system development and rapid implementation.

5.2 ABOUT MACHINE LEARNING

Machine Learning is a branch of Artificial Intelligence that enables systems to learn from data and improve their performance without being explicitly programmed. Instead of following fixed rules, a machine learning system identifies patterns in data and uses them to make predictions or decisions. It is widely used in applications such as recommendation systems, image recognition, speech recognition, medical diagnosis, and automated evaluation systems.

In this project, Machine Learning is used to analyze and evaluate subjective answers automatically. By learning from previously evaluated answers, the system can understand patterns in correct and incorrect responses and generate fair evaluation scores. This reduces human effort and increases consistency in the evaluation process.

Machine Learning improves accuracy over time as more data is provided. It makes the system scalable and capable of handling large volumes of student answers. It also reduces bias and human errors that occur in manual evaluation.

Advantages of Machine Learning:

- Automates decision-making and analysis
- Improves accuracy with more data
- Reduces manual effort and human error
- Handles large datasets efficiently
- Provides intelligent and adaptive systems

Disadvantages of Machine Learning:

- Requires quality data for better results
- Initial training can be time-consuming
- Model performance depends on dataset accuracy
- Requires computational resources

Machine Learning plays a crucial role in making the proposed system intelligent and reliable. It allows the evaluation system to learn from data, adapt to different answer patterns, and provide fair and consistent assessment, making it highly suitable for an AI-based automatic examination paper evaluation system.

5.3 MODULES

A. KEYWORDS

Keywords are specific terms related to a question that are essential for answering it.

These keywords are crucial for determining the score given by the similarity measurement module. They should only include the necessary words in lowercase.

B. SOLUTION

The solution is a subjective answer that is used to compare students' responses.

This solution must include all the keywords and related contexts from the answers, each presented on separate lines or paragraphs. Usually, the teacher or evaluator prepares this solution.

C. ANSWER

The answer is the student's subjective response that needs to be assessed.

It typically includes some or all of the keywords and can range from one to several sentences, depending on the question type and the student's writing style. It often uses synonyms compared to the solution, so careful semantic processing is required during evaluation.

D. DATA COLLECTION

To train and test our model, we need a large set of subjective question-answer data.

However, there is no publicly available labeled dataset for this purpose. In this work, we created our own labeled corpus of subjective answers. To build this corpus, we targeted websites and blogs where such questions and answers are common. We gathered data from various sources across different domains, such as computer science and general knowledge.

E. DATA ANNOTATION

Once the data was collected, it needed labeling.

We selected a group of volunteers from our target domain. We hired 30 annotators from different colleges and universities in various cities in Pakistan. Most of them are students and teachers, with an average age between 21-25, though some are older, ranging from 27-51. These annotators were tasked with scoring the subjective answers based on the students' responses.

F. PREPROCESSING MODULE

After receiving input from the user, both the solution and the answer go through preprocessing.

This involves tokenization, stemming, lemmatization, removing stop words, converting text to lowercase, and adding synonyms. It's important to note that stop words are not removed when using word2vec since it benefits from them for better semantic understanding. However, stop words are removed before inputting data into machine learning models like Multinomial Naive Bayes, as they can interfere with pattern recognition.

G. SIMILARITY MEASUREMENT MODULE

This module includes WDM and Cosine Similarity functions, which are used to calculate the similarity between two sentences or word vectors.

WDM measures dissimilarity, while Cosine Similarity measures similarity. Our approach applies both methods and compares the results at the end. Different similarity and dissimilarity thresholds are used. The WDM thresholds, WDM_LOWER and WDM_UPPER, represent the level of dissimilarity between sentences. A WDM_LOWER threshold of 0.7 was found to indicate very similar sentences, while a WDM_UPPER threshold of 1.6 suggests less similarity. Anything above 1.6 is considered too dissimilar. Cosine similarity thresholds, COS_LOWER and COS_UPPER, are used to measure similarity between sentences. It should be noted that cosine similarity does not account for sentence context, unlike WDM, so both methods are used together.

H. RESULT PREDICTING MODULE

The Result Predicting Module is the central part of this work.

It operates using Algorithm 1. The module calculates an overall score using either WDM or Cosine Similarity, considering the best-matched sentence pairs between the solution and the answer. This result can be compared with an actual score or used to train a machine learning model.

I. MACHINE LEARNING MODEL MODULE

This module consists of machine learning models trained on data from the Result Prediction Module.

Its process is as follows:

• Input data from the Result Prediction Module.

• Preprocess the solution and answer by removing stop words and using Count vectorizer to represent them in either Bag of Words or TF-IDF format.

• Convert the overall score from the Result Prediction Module into a category.

The paper uses four categories (A, B, C, and D), representing the first, second, third, and fourth quarters of a 100. For example, category A represents scores from 0 to 25, and category B represents 26 to 50.

• The number of categories is kept minimal due to the lack of an actual dataset.

These categories could be expanded to cover smaller score ranges.

• A machine learning model like Multinomial Naive Bayes, effective for multi-class classification, is selected.

• The pre-processed answer is used as test data with the model to predict its category, which is then compared with the result from the Result Prediction Module.

This comparison helps build confidence in the model's predicted outcome.

5.4 CODING

File Name: views.py (Admin)

```
from ast import Return

# from ssl import _PasswordType
from django.db.models import Avg,Max,Min,Sum,Count,StdDev,Variance
from django.shortcuts import render,redirect,get_object_or_404
from django.contrib import messages
from adminapp.models import *
from userapp.models import *
from django.core.paginator import Paginator

def admin_index(request):
    pend = UserdetailsModel.objects.filter(user_status="pending").count()
    all = UserdetailsModel.objects.all().count()
    ques = QuestionModel.objects.all().count()
    ans = AnswerModel.objects.all().count()
    return render(request,"admin/admin-index.html",{'ques':ques,'all':all,'ans':ans,'pend':pend})

def admin_pending(request):
    pending=UserdetailsModel.objects.filter(user_status="pending")
    paginator = Paginator(pending,5)
    page_no = request.GET.get('page')
    page = paginator.get_page(page_no)
    return render(request,"admin/admin-pending.html",{'pend':page})

def admin_all(request):
    all=UserdetailsModel.objects.all()
    paginator = Paginator(all,5)
    page_no = request.GET.get('page')
    page = paginator.get_page(page_no)
    return render(request,"admin/admin-all.html",{'all':page})
```

```
def admin_add_subject(request):
    sub =  SubjectModel.objects.all()
    if request.method == "POST" and request.FILES['photo']:
        subject = request.POST.get('subject')
        photo = request.FILES['photo']

        try:
            SubjectModel.objects.get(subject = subject.lower())

            messages.error(request,"This subject already exists, Try another subject")
            return redirect('admin_add_subject')
        except:
            SubjectModel.objects.create(subject = subject.lower(),subject_image = photo)
            messages.success(request,subject+" subject added successfully")
            return redirect('admin_add_subject')
    return render(request,"admin/admin-add-subject.html",{'sub':sub})


def admin_add_question(request):
    sub = SubjectModel.objects.all()
    print(sub,'sub model')
    if request.method == "POST":
        question = request.POST.get('question')
        answer = request.POST.get('answer')
        subject = request.POST.get('subject')
        if QuestionModel.objects.filter(subject = subject).count() <= 4:

            QuestionModel.objects.create(question = question,answer = answer,subject = subject)
            messages.success(request,"Question added successfully")
            return redirect('admin_add_question')
        else:
            messages.info(request,"Limit reached for question in "+subject+", Remove some questions
to add new questions")
            return redirect('admin_add_question')
    return render(request,"admin/admin-add-question.html",{'sub':sub})
```

```
def admin_manage_question(request):
    ques = QuestionModel.objects.all()
    paginator = Paginator(ques,5)
    page_no = request.GET.get('page')
    page = paginator.get_page(page_no)
    return render(request,"admin/admin-manage-question.html",{'ques':page})

def admin_results(request):
    result = AnswerModel.objects.all()
    paginator = Paginator(result,5)
    page_no = request.GET.get('page')
    page = paginator.get_page(page_no)
    return render(request,"admin/admin-results.html",{'result':page})

def admin_analysis_graph(request):
    f = AnswerModel.objects.filter(grade = 'F').count()
    c = AnswerModel.objects.filter(grade = 'C').count()
    b = AnswerModel.objects.filter(grade = 'B').count()
    a = AnswerModel.objects.filter(grade = 'A').count()

    return render(request,"admin/admin-analysis-graph.html",{'a':a,'b':b,'c':c,'f':f,})

def accept_user(request,user_id):
    accept = get_object_or_404(UserdetailsModel,user_id=user_id)
    accept.user_status = "accepted"
    accept.save(update_fields=["user_status"])
    accept.save()
    if accept:
        messages.success(request,"User Added Successfully")

    return redirect('admin_pending')

def decline_user(request,user_id):
    decline = get_object_or_404(UserdetailsModel,user_id=user_id)
    decline.user_status = "declined"
```

```
    decline.save(update_fields=["user_status"])

 decline.save()
    if decline:
        messages.success(request,"Rejected Successfully")

    return redirect('admin_pending',user_id)

def remove_questions(request,question_id):
    remove = get_object_or_404(QuestionModel,question_id=question_id).delete()

    if remove:
        messages.success(request,"Question Removed Successfully")
    return redirect('admin_manage_question')

def remove_subject(request,subject_id):
    remove = get_object_or_404(SubjectModel,subject_id=subject_id).delete()
    if remove:
        messages.success(request,"Subject Removed Successfully")
    return redirect('admin_add_subject')
```

File Name: views.py (User)

```
from django.db.models import Avg,Max,Min,Sum,Count,StdDev,Variance
from django.shortcuts import render,redirect,get_object_or_404
from django.contrib import messages
from adminapp.models import *
from userapp.models import *
from django.core.paginator import Paginator
from userapp.text_similarity import *
import nltk
nltk.download('stopwords')
nltk.download('punkt')
import json
import ast
```

```
# Create your views here.
def index(request):
    return render(request,"user/index.html")

def admin_login(request):
    if request.method == "POST":
        username=request.POST.get("username")
        password=request.POST.get("password")

        if username == "admin" and password == "admin":
            messages.success(request, "Logged In Successfully.")
            return redirect('admin_index')
        else:
            messages.error(request,"Invalid Username or Password")
            return redirect('admin_login')
    return render(request,"admin/admin-login.html")

def user_login(request):
    if request.method == "POST":
        username=request.POST.get("email")
        password=request.POST.get("password")

        try:
            auth = UserdetailsModel.objects.get(user_email=username,user_password=password)
            if auth.user_status  == "accepted":
                request.session['user_id'] = auth.user_id
                messages.success(request,'Successfully Logged In')
                return redirect('user_dashboard')
            elif auth.user_status == "pending":
                messages.info(request,'Your id is pending for registration ')
                return redirect('user_login')
            elif auth.user_status == "blocked":
                messages.error(request,'You Are BLOCKED From Logging In ')
                return redirect('user_login')
            else:
```

```
                messages.error(request,'You are not registered,try again after signup')
                return redirect('user_login')

        except:
            messages.error(request,'invalid login credentials')
            return redirect('user_login')
    return render(request,"user/user-login.html")

def user_register(request):
    if request.method == "POST" and request.FILES["photo"] :
        name = request.POST['name']
        email = request.POST.get('email')
        contact = request.POST.get('contact')
        password = request.POST.get('password')
        id = request.POST.get('id')
        photo = request.FILES['photo']

        try:
            a = UserdetailsModel.objects.get(user_email=email)
            messages.info(request,"Email already exists,try again with another email")
            return redirect('user_register')
        except:
            user_create = UserdetailsModel.objects.create(user_name=name, user_email=email,
user_password=password, user_contact=contact,student_id=id,user_photo=photo)
            if user_create:
                messages.success(request,"Successfully Registered")
                return redirect('user_register')
            else:
                messages.error(request,"invalid details, try again")
                return redirect('user_register')
    return render(request,"user/user-register.html")

def user_contact(request):
    return render(request,"user/user-contact.html")
```

```
def user_dashboard(request):
    user_id=request.session['user_id']
    user=UserdetailsModel.objects.get(user_id=user_id)

 exam  = AnswerModel.objects.filter(user_id=user_id).count()
    exams  = AnswerModel.objects.filter(user_id=user_id)
    sub = []
    for i in exams:
        if i.answer_subject not in sub:
            sub.append(i.answer_subject)
    ques = AnswerModel.objects.filter(user_id=user_id).count()*5
    su = len(sub)
    return render(request,"user/user-dashboard.html",{'exams':exam,'subjects':su,'ques':ques})

def user_questions(request,subject):
    user_id=request.session['user_id']
    user=UserdetailsModel.objects.get(user_id=user_id)

    sub = QuestionModel.objects.filter(subject = subject)
    for i in sub:
        pass
    if request.method == 'POST':
        q1 = request.POST.get('question1')
        q2 = request.POST.get('question3')
        q3 = request.POST.get('question4')
        q4 = request.POST.get('question5')
        q5 = request.POST.get('question6')

        r1 = int((text_similarity_nltk(q1,sub[0].answer))*20)
        r2 = int((text_similarity_nltk(q2,sub[1].answer))*20)
        r3 = int((text_similarity_nltk(q3,sub[2].answer))*20)
        r4 = int((text_similarity_nltk(q4,sub[3].answer))*20)
        r5 = int((text_similarity_nltk(q5,sub[4].answer))*20)

        score = r1+r2+r3+r4+r5
```

```
        grade =''
        if score == 0 or score <= 24:
            grade='F'
        elif score >= 25 and score <= 49:
            grade = 'C'

elif score >=50 and score <= 75:
            grade = 'B'
        elif score >= 76 and score <= 100:
            grade = 'A'

        answer = {}
        answer['question1'] ={'question':sub[0].question,'answer':q1,'marks':str(r1)}
        answer['question2'] ={'question':sub[1].question,'answer':q2,'marks':str(r2)}
        answer['question3'] ={'question':sub[2].question,'answer':q3,'marks':str(r3)}
        answer['question4'] ={'question':sub[3].question,'answer':q4,'marks':str(r4)}
        answer['question5'] ={'question':sub[4].question,'answer':q5,'marks':str(r5)}

        AnswerModel.objects.create(answer_subject = subject,answer = answer,user_id = user,score
= score,grade = grade)

        messages.success(request,"Answers submitted successfully")
        return redirect("user_exam")
    try:
        return render(request,"user/user-
questions.html",{'quest1':sub[0],'quest2':sub[1],'quest3':sub[2],'quest4':sub[3],'quest5':sub[4],})
    except:
        messages.error(request,"Something Went Wrong, Check with admin if questions are added
and try again.")
        return redirect("user_exam")

def user_view_results(request,answer_id):
    TempModel.objects.all().delete()
    result = AnswerModel.objects.get(answer_id = answer_id)
    dict = result.answer
```

```
    str_to_dict= ast.literal_eval(dict)

    for key ,value in str_to_dict.items():
        TempModel.objects.create(subject = result.answer_subject,question =
value['question'],answer = value['answer'],score = value['marks'])

    f_results =  TempModel.objects.all()

    return render(request,"user/user-view-results.html",{'result':f_results})

def user_results(request):
    user_id=request.session['user_id']

    result = AnswerModel.objects.filter(user_id = user_id)
    return render(request,"user/user-results.html",{'result':result})

def user_myprofile(request):
    user_id=request.session['user_id']
    user=UserdetailsModel.objects.get(user_id=user_id)

    if request.method=="POST":
        if len(request.FILES) ==0:
            name=request.POST.get("name")
            email=request.POST.get("email")
            password=request.POST.get("password")
            contact=request.POST.get("contact")
            id = request.POST.get("id")

            user.user_name = name

            user.user_email = email
            user.user_password = password
            user.user_contact = contact
            user.student_id =  id
            user.save()
```

```
            if user:
                messages.success(request,"Successfully Updated")
                return redirect("user_myprofile")
            else:
                messages.error(request,"No changes detected")
                return redirect("user_myprofile")

    else:
        if request.method=="POST" and request.FILES['photo']:
            name=request.POST.get("name")
            email=request.POST.get("email")
            password=request.POST.get("password")
            contact=request.POST.get("contact")
            id = request.POST.get("id")

            photo=request.FILES["photo"]
            user.user_name = name
            user.user_email = email
            user.user_password = password
            user.user_contact = contact
            user.student_id =  id
            user.user_photo = photo
            user.save()
            if user:
                messages.success(request,"Successfully Updated")
                return redirect("user_myprofile")
            else:
                messages.error(request,"No changes detected")
                return redirect("user_myprofile")
        return render(request,"user/user-myprofile.html",{'user':user})

    def user_exam(request):
        sub = SubjectModel.objects.all()

        return render(request,"user/user-exam.html",{'sub':sub})
```

6. SYSTEM TESTING

System testing is performed to ensure that the developed application works correctly and meets the specified requirements. It verifies the correctness of individual modules as well as their interaction with each other. In this project, system testing was carried out at different levels to ensure reliability, accuracy, and proper functioning of the AI-based Automatic Examination Paper Evaluation System.

The following testing methods were used:

6.1 UNIT TESTING

Unit testing is performed on individual modules of the system to verify their correctness. Each module was tested separately to ensure that it produces the expected output for a given input.

In this project, unit testing was carried out on modules such as:

- Text preprocessing
- Similarity measurement
- Keyword matching
- Result prediction

By testing each unit independently, errors were detected at an early stage, which helped improve the overall stability of the system.

6.2 INTEGRATION TESTING

Integration testing is performed after combining individual modules of the system. The objective of this testing is to verify proper data flow and interaction between different modules.

In the proposed system, integration testing ensured that:

- Student answers are correctly passed to the preprocessing module
- Preprocessed data is correctly sent to the similarity measurement module
- Similarity results are properly used for final score generation
- Backend logic interacts correctly with the database and frontend

This testing confirmed that all integrated modules work together without errors.

6.3 FUNCTIONAL TESTING

Functional testing is performed to verify that all features of the system work according to the specified requirements. It focuses on validating the system's functionality rather than internal implementation.

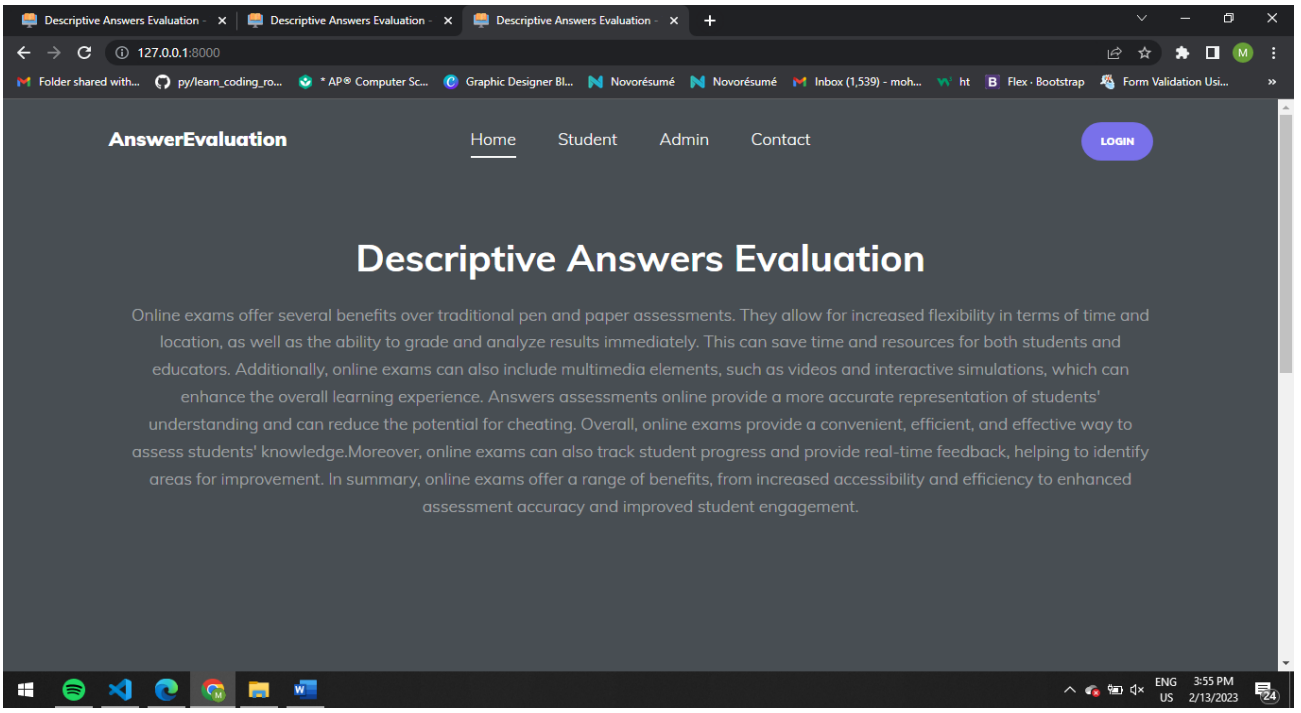
In this project, functional testing was conducted to check:

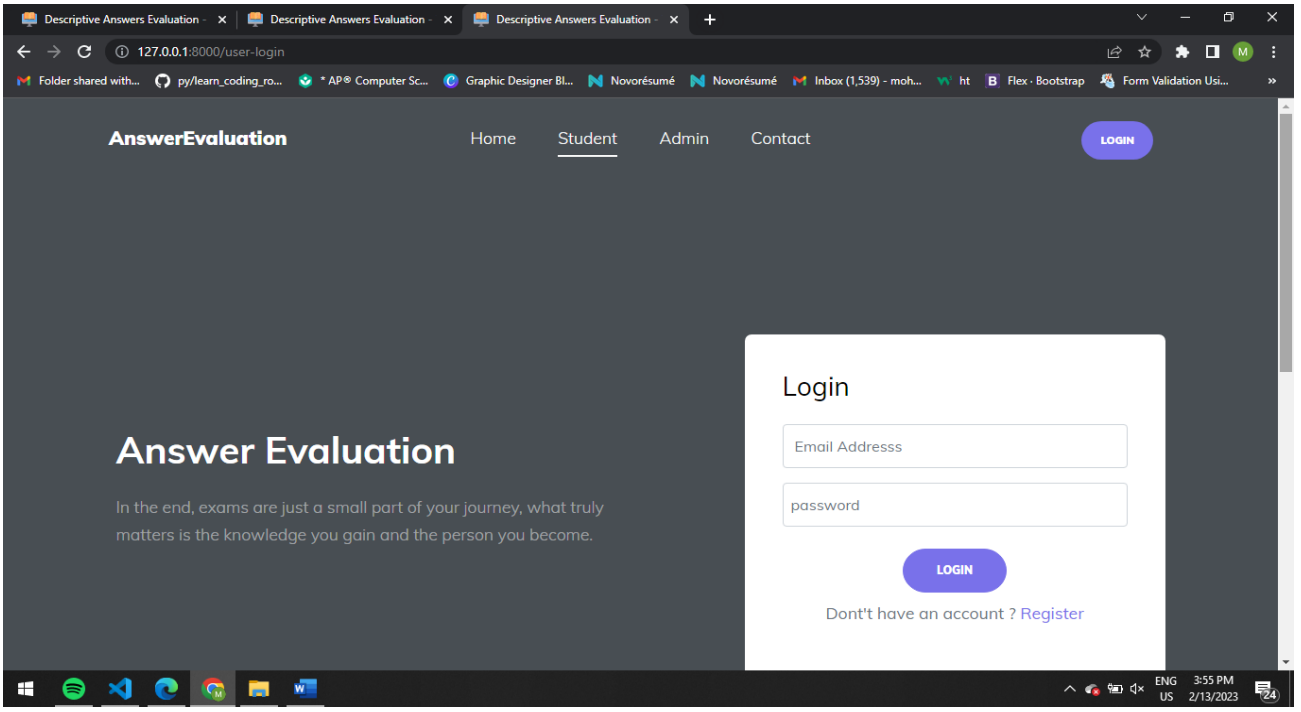
- User login and authentication
- Question, solution, and keyword input
- Student answer submission
- Automatic evaluation and score calculation
- Display of results

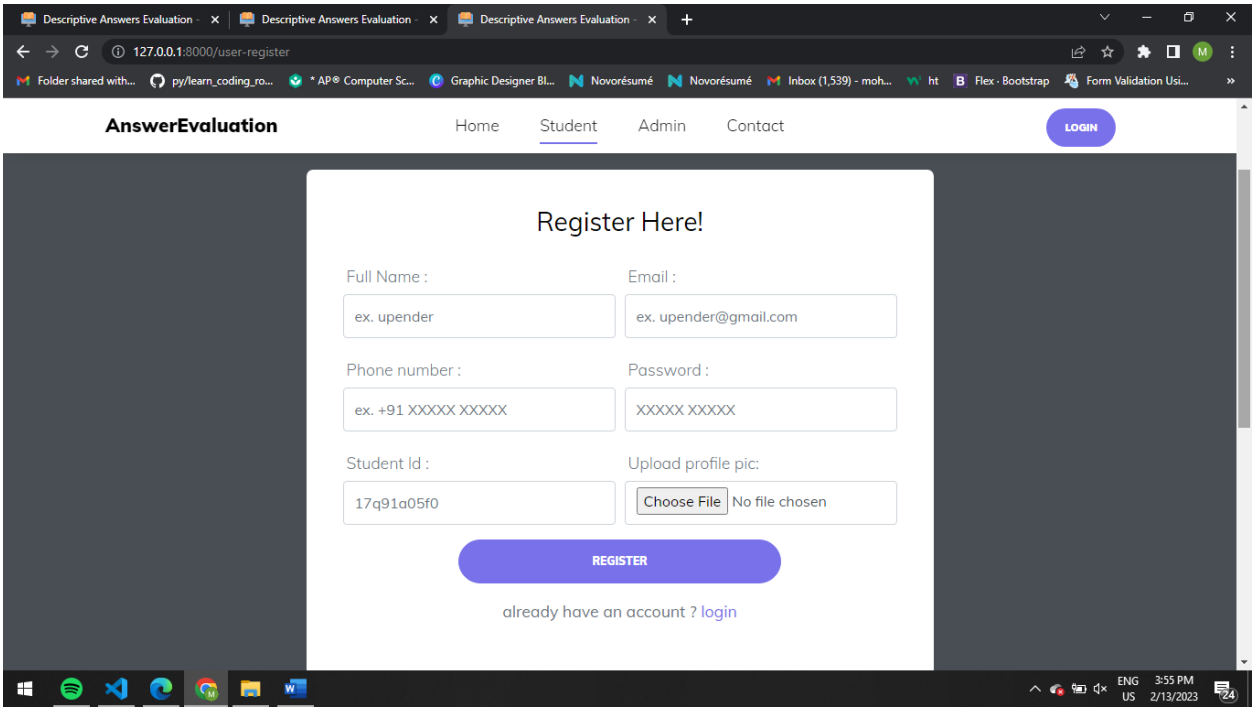
Functional testing ensured that the system behaves as expected for valid inputs and provides accurate evaluation results.

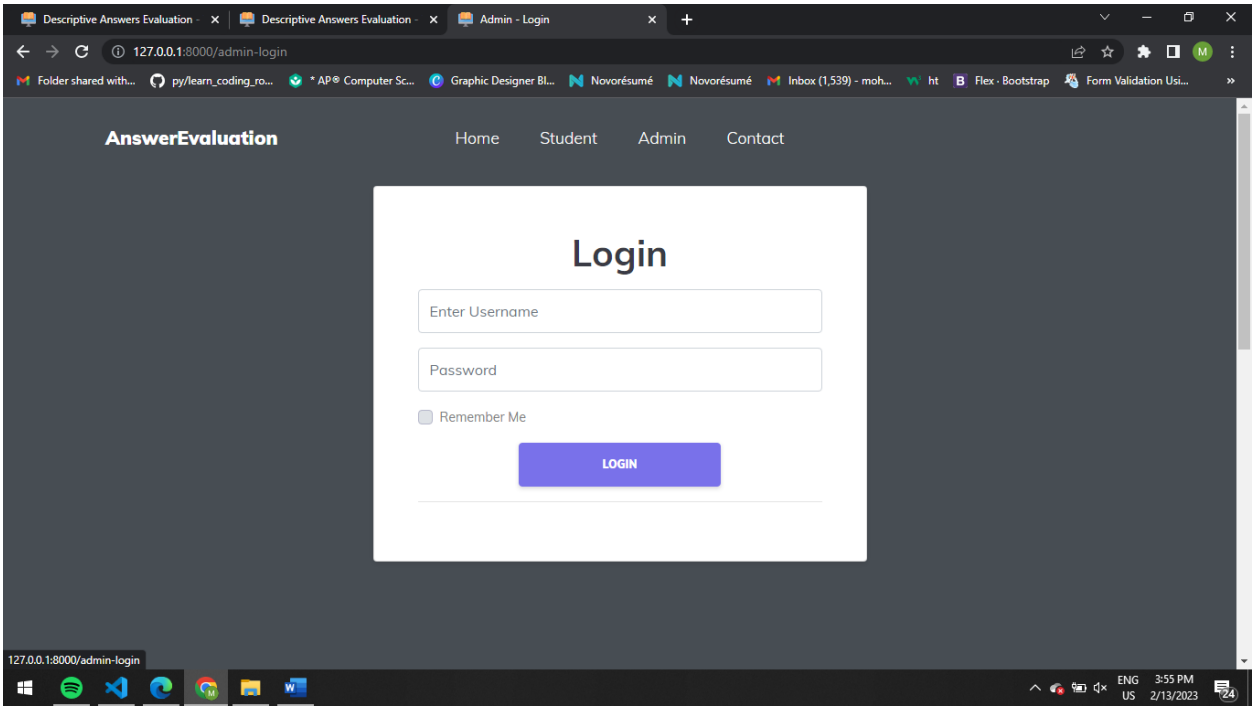
7. SCREENSHOTS

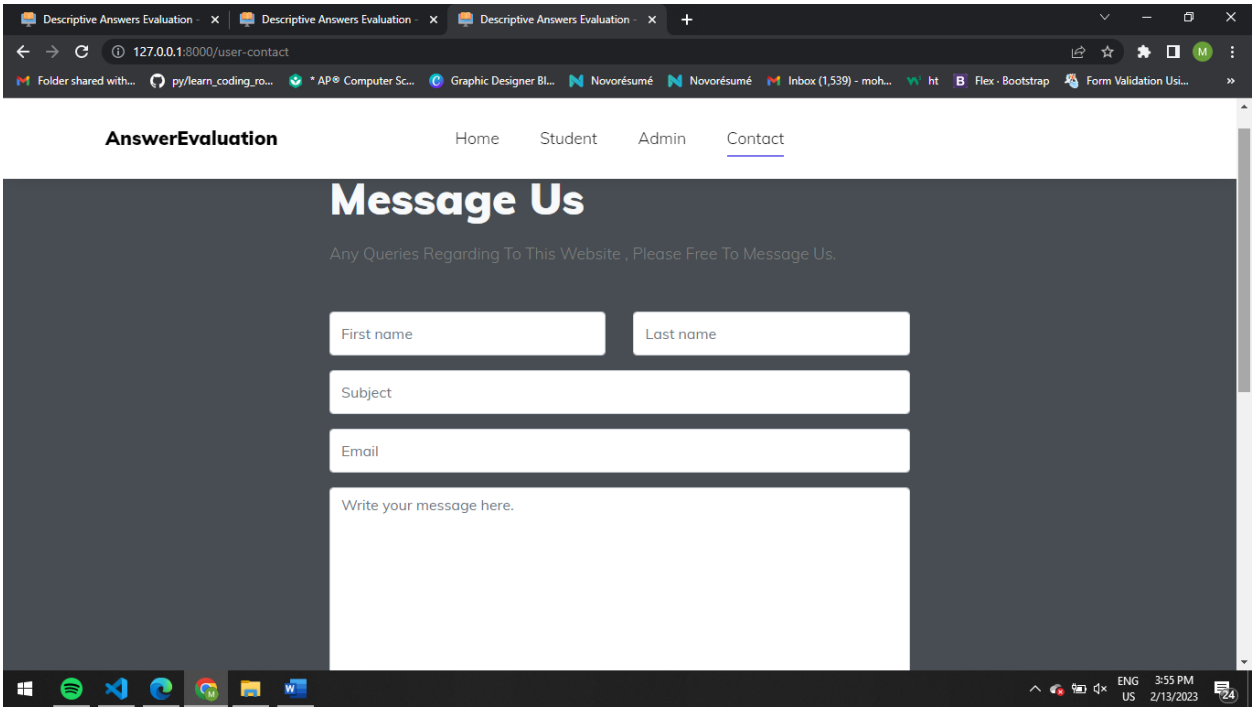
MAIN PAGE:



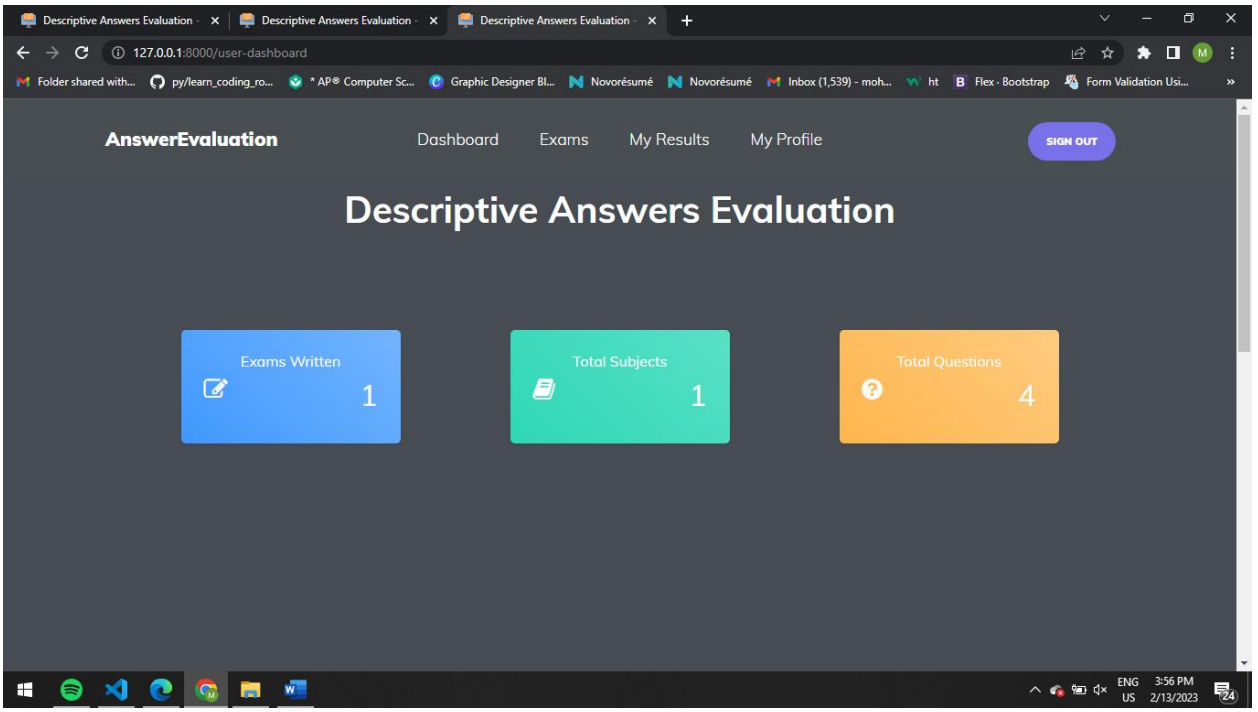


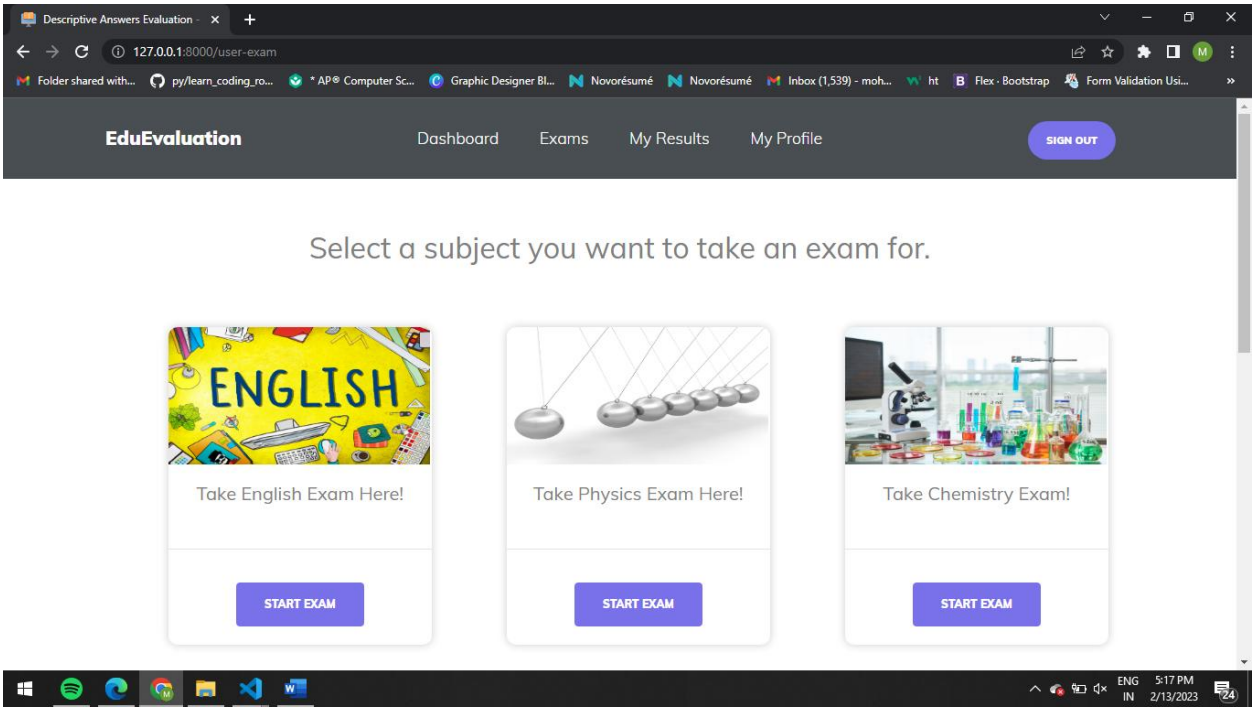

Descriptive Answers Evaluation
127.0.0.1:8000/user-register
AnswerEvaluation
Home
Student
Admin
Contact
LOGIN
Register Here!
Full Name :
ex. upender
Email :
ex. upender@gmail.com
Phone number :
ex. +91 XXXXX XXXXX
Password :
XXXXX XXXXX
Student Id :
17q91a05f0
Upload profile pic:
Choose File
No file chosen
REGISTER
already have an account ? login

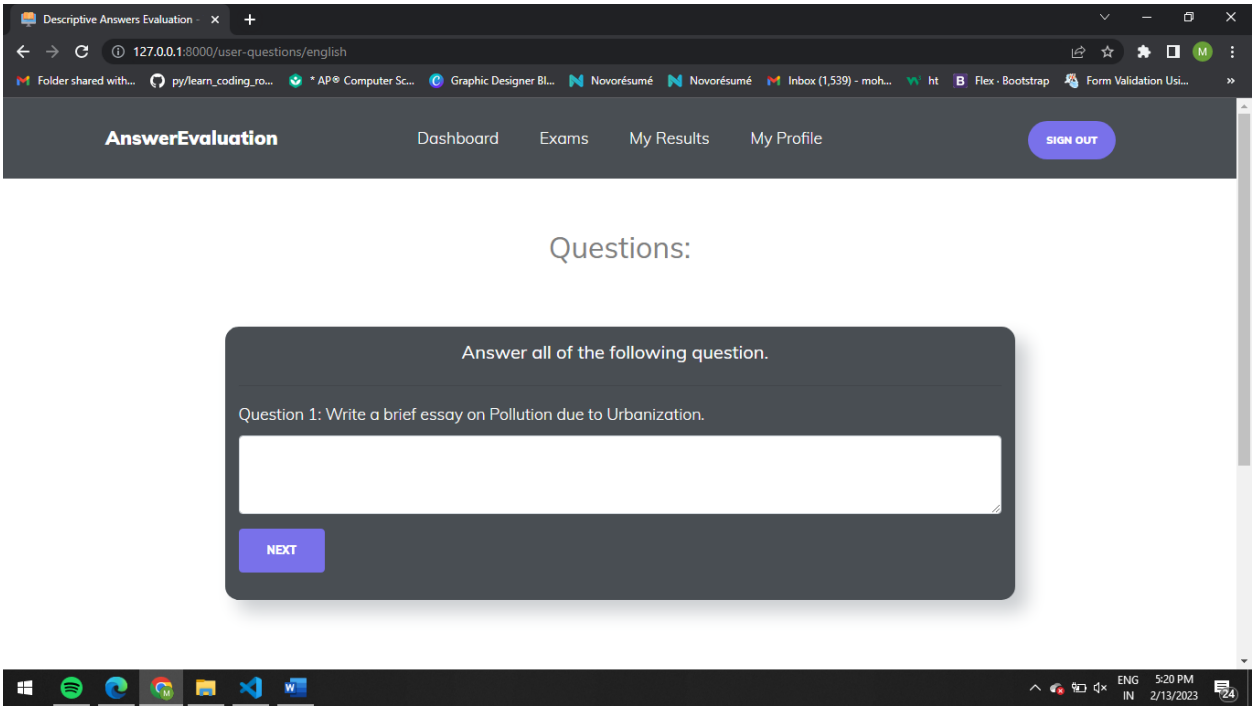

Descriptive Answers Evaluation
Admin - Login
127.0.0.1:8000/admin-login
AnswerEvaluation
Home
Student
Admin
Contact
Login
Enter Username
Password
Remember Me
LOGIN
127.0.0.1:8000/admin-login

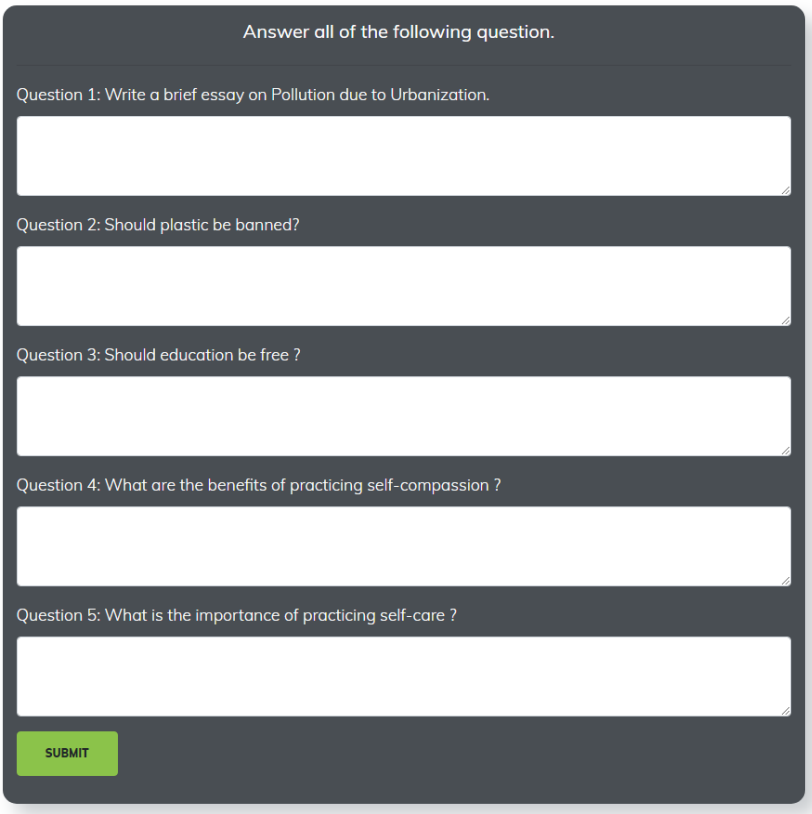

Descriptive Answers Evaluation
127.0.0.1:8000/user-contact
AnswerEvaluation
Home
Student
Admin
Contact
Message Us
Any Queries Regarding To This Website , Please Free To Message Us.
First name
Last name
Subject
Email
Write your message here.

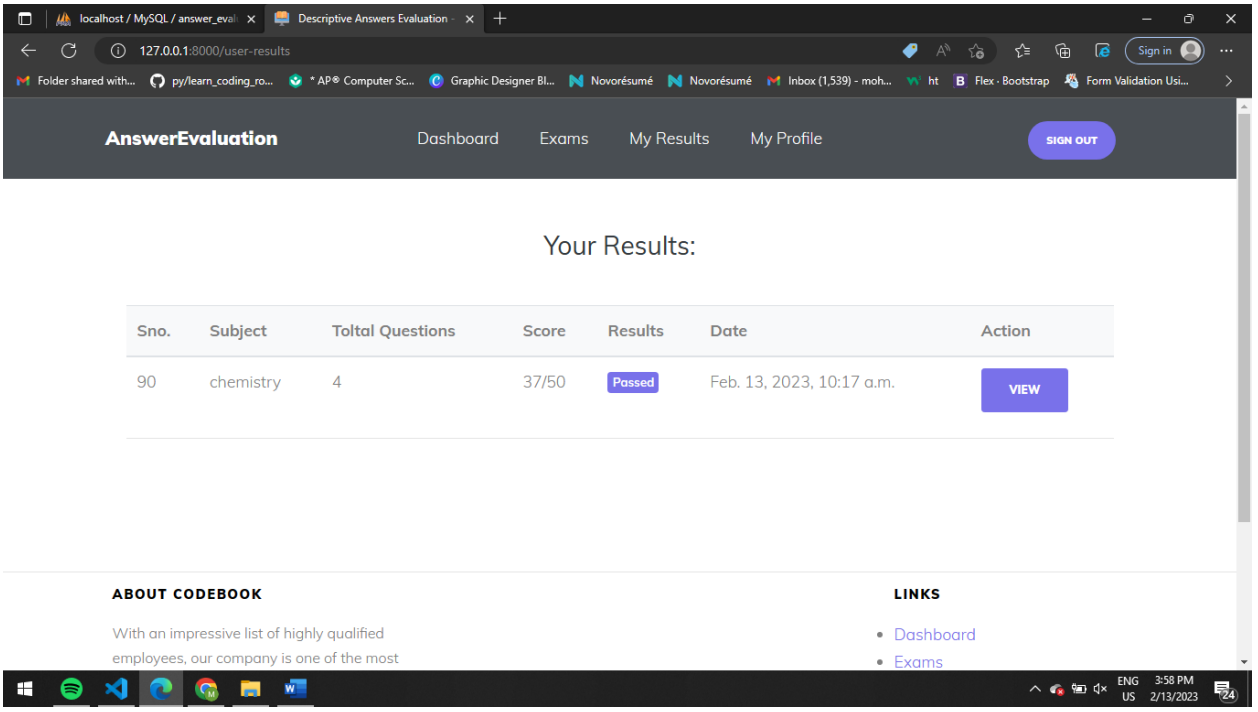
USER PAGE:

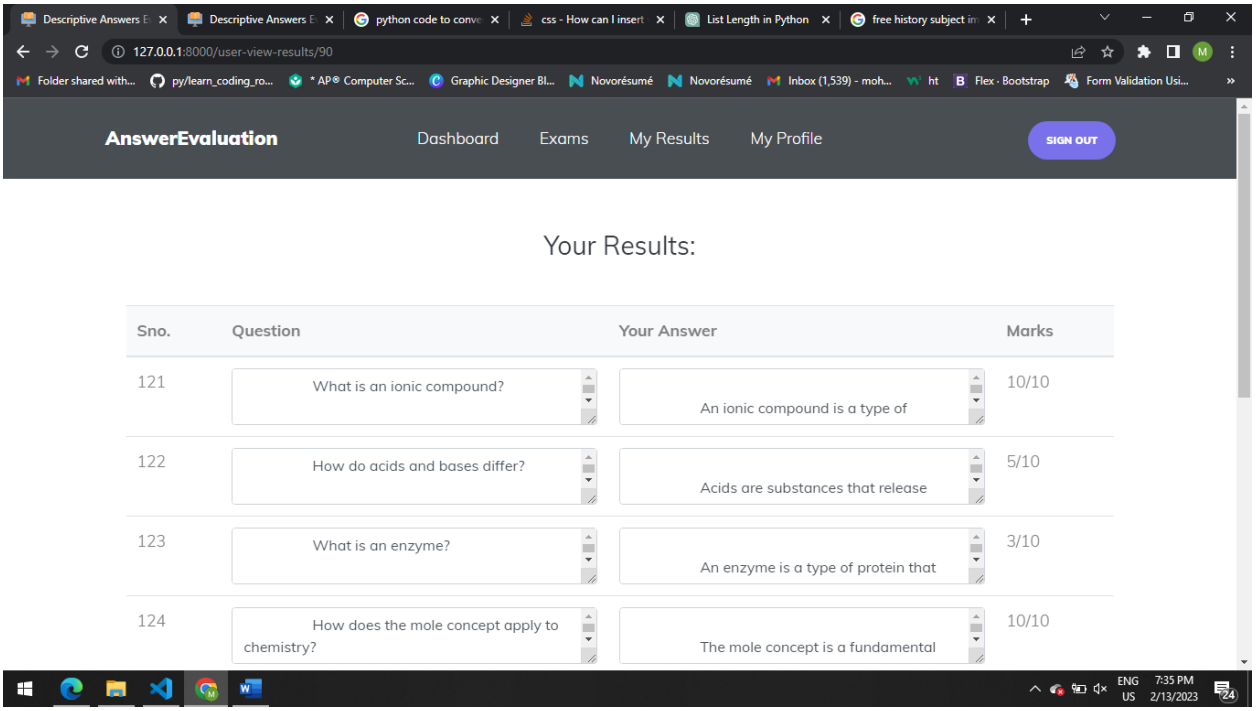

Descriptive Answers Evaluation
127.0.0.1:8000/user-dashboard
AnswerEvaluation
Dashboard
Exams
My Results
My Profile
SIGN OUT
Descriptive Answers Evaluation
Exams Written
1
Total Subjects
1
Total Questions
4

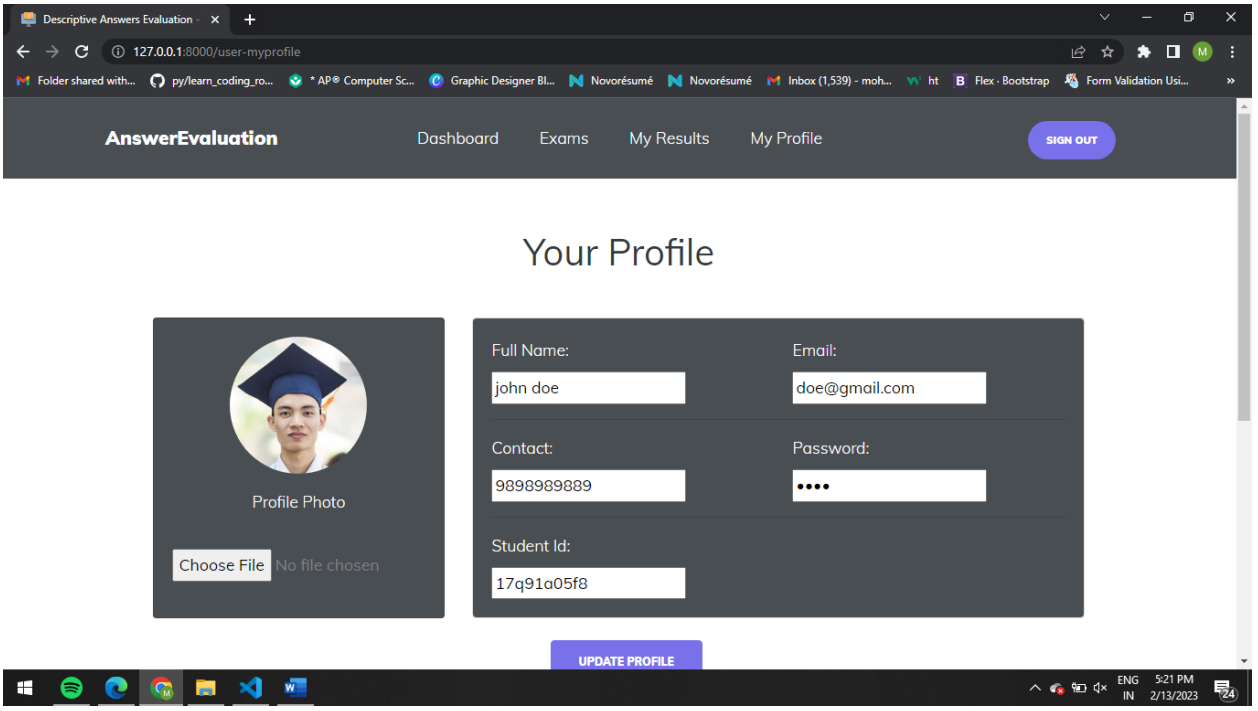

Descriptive Answers Evaluation
127.0.0.1:8000/user-exam
EduEvaluation
Dashboard
Exams
My Results
My Profile
SIGN OUT
Select a subject you want to take an exam for.
ENGLISH
Take English Exam Here!
START EXAM
Take Physics Exam Here!
START EXAM
Take Chemistry Exam!
START EXAM


Descriptive Answers Evaluation
127.0.0.1:8000/user-questions/english
AnswerEvaluation
Dashboard
Exams
My Results
My Profile
SIGN OUT
Questions:
Answer all of the following question.
Question 1: Write a brief essay on Pollution due to Urbanization.
NEXT

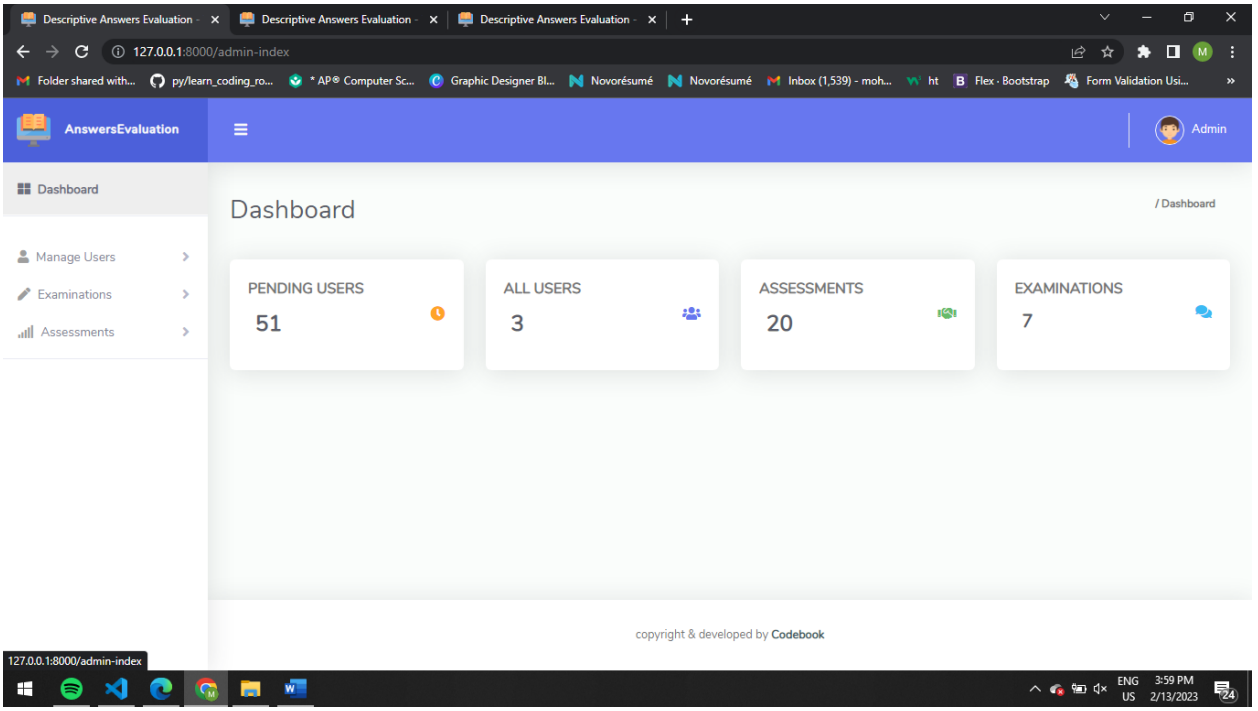

Answer all of the following question.
Question 1: Write a brief essay on Pollution due to Urbanization.
Question 2: Should plastic be banned?
Question 3: Should education be free ?
Question 4: What are the benefits of practicing self-compassion ?
Question 5: What is the importance of practicing self-care ?
SUBMIT

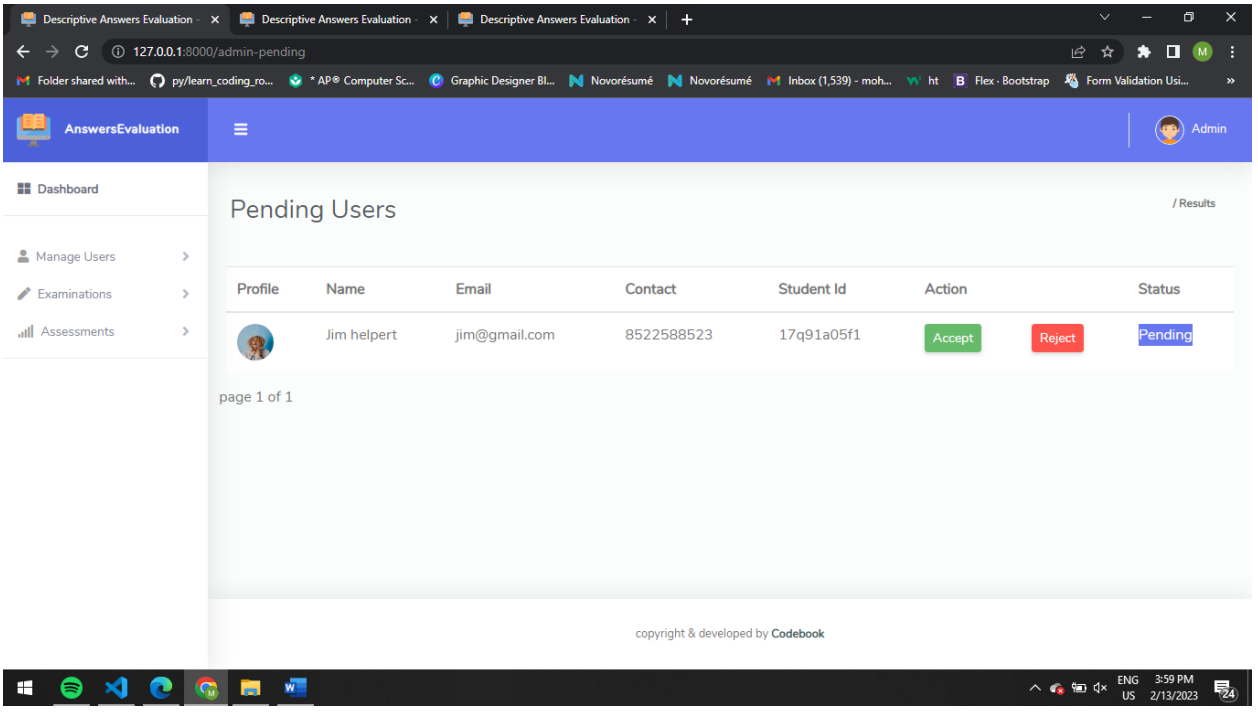

localhost / MySQL / answer_eval
Descriptive Answers Evaluation
127.0.0.1:8000/user-results
AnswerEvaluation
Dashboard
Exams
My Results
My Profile
SIGN OUT
Your Results:
Sno.
Subject
Toltal Questions
Score
Results
Date
Action
90
chemistry
4
37/50
Passed
Feb. 13, 2023, 10:17 a.m.
VIEW
ABOUT CODEBOOK
With an impressive list of highly qualified employees, our company is one of the most
LINKS
Dashboard
Exams

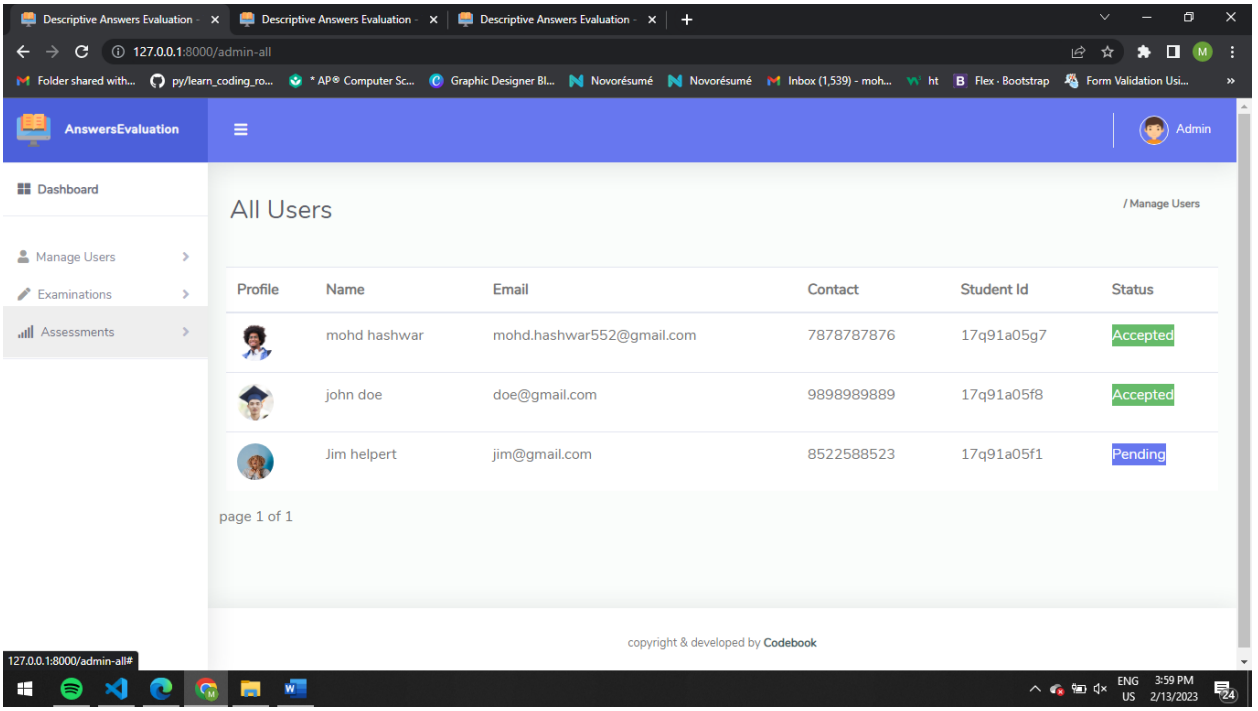

AnswerEvaluation
Dashboard
Exams
My Results
My Profile
SIGN OUT
Your Results:
Sno.
Question
Your Answer
Marks
121
What is an ionic compound?
An ionic compound is a type of
10/10
122
How do acids and bases differ?
Acids are substances that release
5/10
123
What is an enzyme?
An enzyme is a type of protein that
3/10
124
How does the mole concept apply to chemistry?
The mole concept is a fundamental
10/10

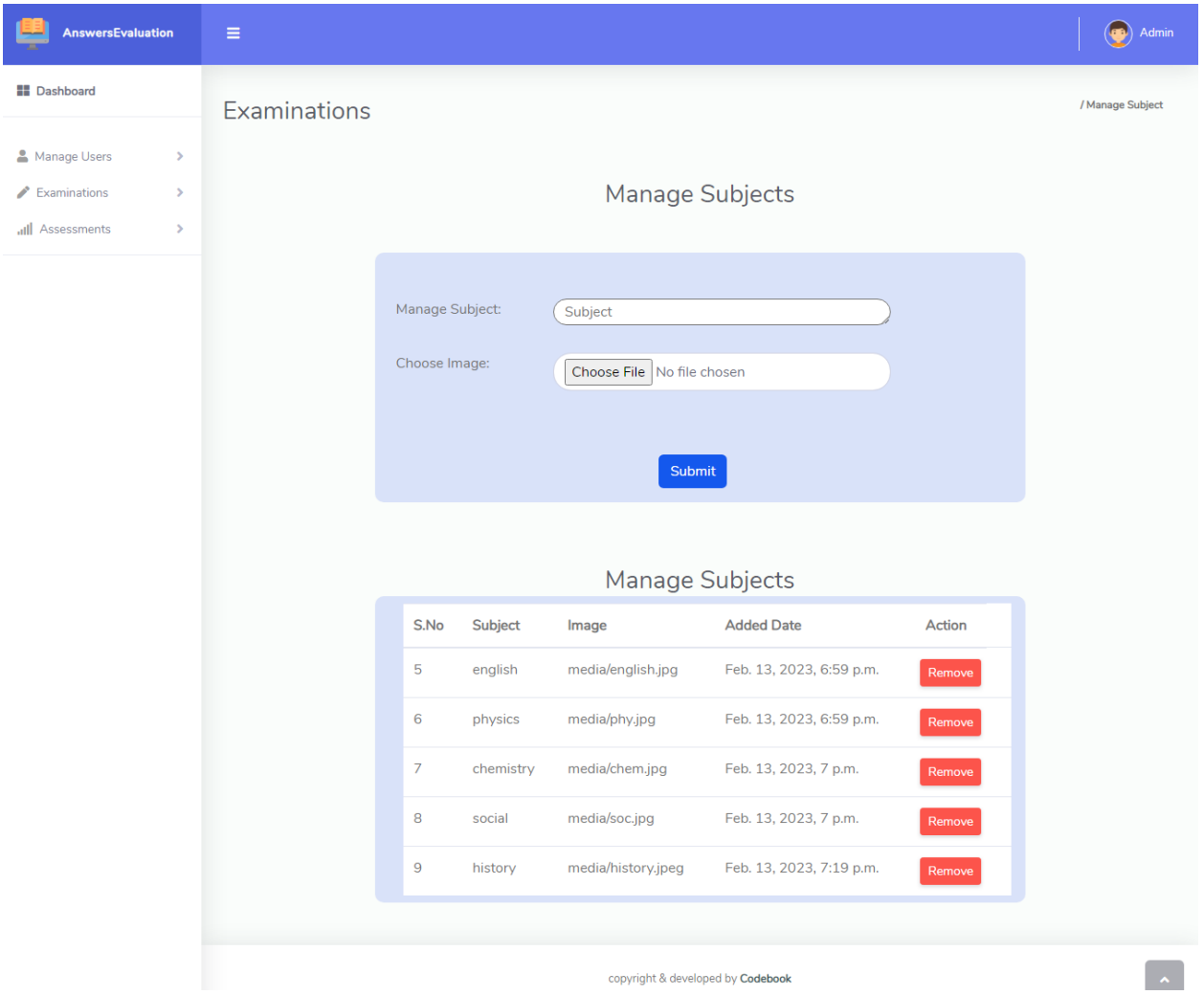

AnswerEvaluation
Dashboard
Exams
My Results
My Profile
SIGN OUT
Your Profile
Profile Photo
Choose File
No file chosen
Full Name:
john doe
Email:
doe@gmail.com
Contact:
9898989889
Password:
Student Id:
17q91a05f8
UPDATE PROFILE

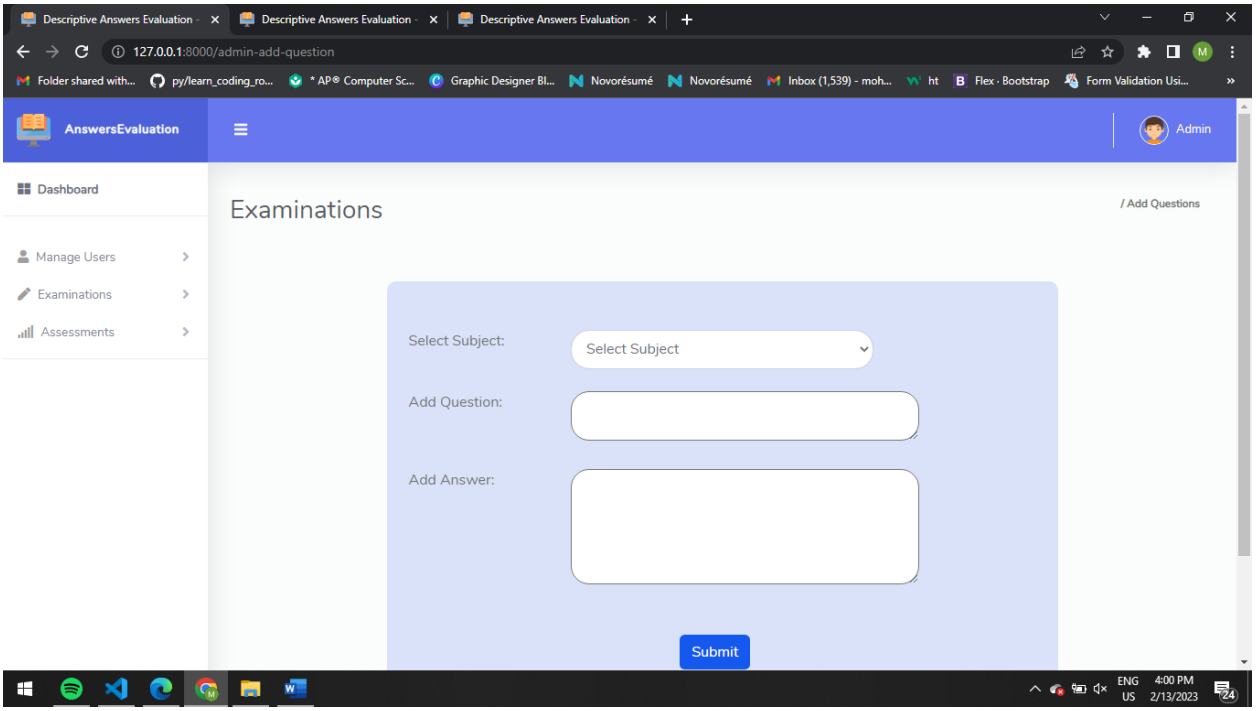
ADMIN PAGE:

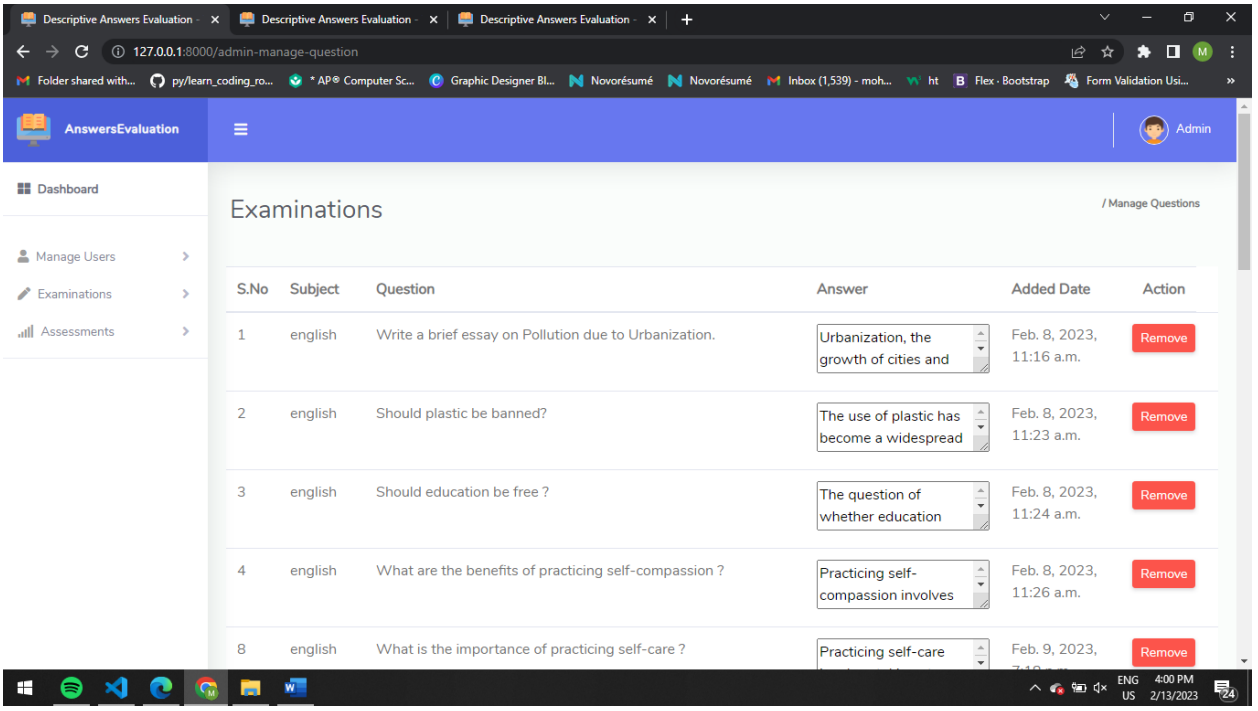

Descriptive Answers Evaluation
127.0.0.1:8000/admin-index
AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
Dashboard
/ Dashboard
PENDING USERS
51
ALL USERS
3
ASSESSMENTS
20
EXAMINATIONS
7
copyright & developed by Codebook
127.0.0.1:8000/admin-index

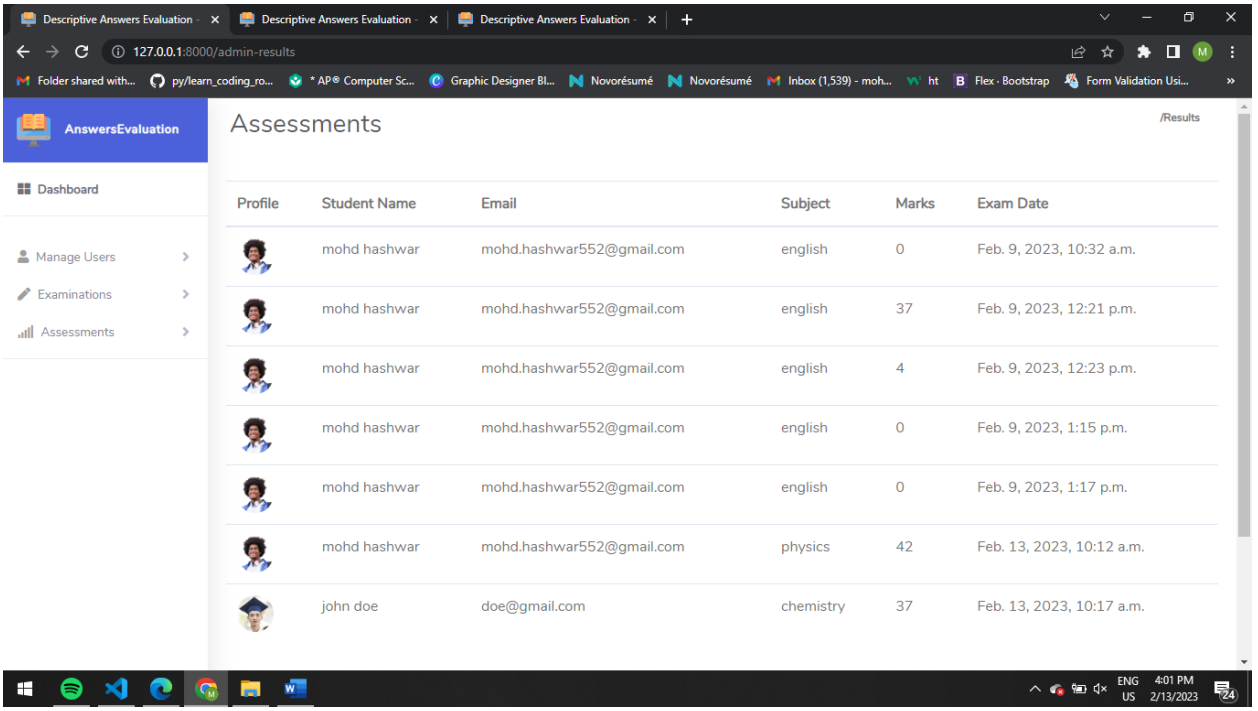

Descriptive Answers Evaluation
127.0.0.1:8000/admin-pending
AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
Pending Users
/ Results
Profile
Name
Email
Contact
Student Id
Action
Status
Jim helpert
jim@gmail.com
8522588523
17q91a05f1
Accept
Reject
Pending
page 1 of 1
copyright & developed by Codebook

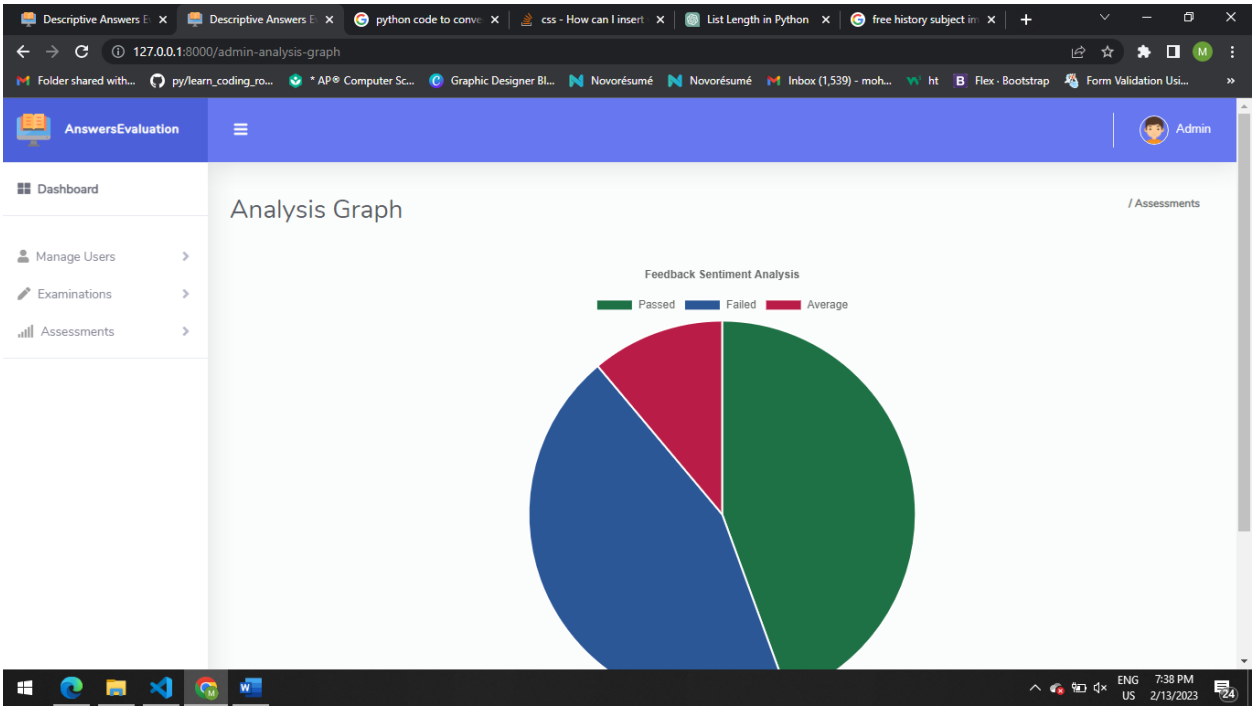

Descriptive Answers Evaluation
127.0.0.1:8000/admin-all
AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
All Users
/ Manage Users
Profile
Name
Email
Contact
Student Id
Status
mohd hashwar
mohd.hashwar552@gmail.com
7878787876
17q91a05g7
Accepted
john doe
doe@gmail.com
9898989889
17q91a05f8
Accepted
Jim helpert
jim@gmail.com
8522588523
17q91a05f1
Pending
page 1 of 1
copyright & developed by Codebook
127.0.0.1:8000/admin-all#
ENG
US
3:59 PM
2/13/2023


AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
Examinations
/ Manage Subject
Manage Subjects
Manage Subject:
Subject
Choose Image:
Choose File
No file chosen
Submit
Manage Subjects
S.No
Subject
Image
Added Date
Action
5
english
media/english.jpg
Feb. 13, 2023, 6:59 p.m.
Remove
6
physics
media/phy.jpg
Feb. 13, 2023, 6:59 p.m.
Remove
7
chemistry
media/chem.jpg
Feb. 13, 2023, 7 p.m.
Remove
8
social
media/soc.jpg
Feb. 13, 2023, 7 p.m.
Remove
9
history
media/history.jpeg
Feb. 13, 2023, 7:19 p.m.
Remove
copyright & developed by Codebook


Descriptive Answers Evaluation
127.0.0.1:8000/admin-add-question
AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
Examinations
/ Add Questions
Select Subject:
Select Subject
Add Question:
Add Answer:
Submit


Descriptive Answers Evaluation
127.0.0.1:8000/admin-manage-question
AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
Examinations
/ Manage Questions
S.No
Subject
Question
Answer
Added Date
Action
1
english
Write a brief essay on Pollution due to Urbanization.
Urbanization, the growth of cities and
Feb. 8, 2023, 11:16 a.m.
Remove
2
english
Should plastic be banned?
The use of plastic has become a widespread
Feb. 8, 2023, 11:23 a.m.
Remove
3
english
Should education be free ?
The question of whether education
Feb. 8, 2023, 11:24 a.m.
Remove
4
english
What are the benefits of practicing self-compassion ?
Practicing self-compassion involves
Feb. 8, 2023, 11:26 a.m.
Remove
8
english
What is the importance of practicing self-care ?
Practicing self-care
Feb. 9, 2023,
Remove


Descriptive Answers Evaluation
127.0.0.1:8000/admin-results
AnswersEvaluation
Dashboard
Manage Users
Examinations
Assessments
Assessments
/Results
Profile
Student Name
Email
Subject
Marks
Exam Date
mohd hashwar
mohd.hashwar552@gmail.com
english
0
Feb. 9, 2023, 10:32 a.m.
mohd hashwar
mohd.hashwar552@gmail.com
english
37
Feb. 9, 2023, 12:21 p.m.
mohd hashwar
mohd.hashwar552@gmail.com
english
4
Feb. 9, 2023, 12:23 p.m.
mohd hashwar
mohd.hashwar552@gmail.com
english
0
Feb. 9, 2023, 1:15 p.m.
mohd hashwar
mohd.hashwar552@gmail.com
english
0
Feb. 9, 2023, 1:17 p.m.
mohd hashwar
mohd.hashwar552@gmail.com
physics
42
Feb. 13, 2023, 10:12 a.m.
john doe
doe@gmail.com
chemistry
37
Feb. 13, 2023, 10:17 a.m.
ENG
US
4:01 PM
2/13/2023


Descriptive Answers E
Descriptive Answers E
python code to conve
css - How can I insert
List Length in Python
free history subject im
127.0.0.1:8000/admin-analysis-graph
AnswersEvaluation
Admin
Dashboard
Manage Users
Examinations
Assessments
Analysis Graph
/ Assessments
Feedback Sentiment Analysis
Passed
Failed
Average
ENG
US
7:38 PM
2/13/2023

8. CONCLUSION

This project successfully presents an **AI-Based Automatic Examination Paper Evaluation System** that automates the evaluation of descriptive answers using Machine Learning and Natural Language Processing techniques. The proposed system addresses the limitations of traditional manual evaluation by providing a faster, more consistent, and unbiased assessment mechanism for subjective answers.

The system effectively processes student responses through multiple stages, including text preprocessing, semantic similarity measurement, keyword analysis, and machine learning-based prediction. By combining contextual similarity techniques with keyword matching, the system ensures fair evaluation even when answers vary in structure, length, or wording. This approach allows the system to handle synonyms and sentence reordering while still preserving the intended meaning of the answer.

The integration of Machine Learning enables the system to improve over time by learning from previously evaluated answers. This reduces dependency on predefined solutions and keywords, making the system scalable and adaptable to different subjects and question patterns. The use of Python and Django provides a flexible and robust framework for developing a web-based evaluation platform, while MySQL ensures secure and efficient data storage.

The system was thoroughly tested using Unit Testing, Integration Testing, and Functional Testing to ensure reliability and correctness. The results demonstrate that the proposed system can accurately evaluate descriptive answers and generate meaningful scores with minimal human intervention. By automating the evaluation process, the system significantly reduces the workload of teachers and minimizes human errors and bias.

Overall, this project proves that intelligent systems can play a vital role in modernizing the education sector. The AI-based evaluation system offers an efficient, transparent, and scalable solution for assessing descriptive answers, making it suitable for academic institutions and online examination platforms. With further enhancements, the system has the potential to become a complete automated assessment tool for future educational needs.

9. FUTURE SCOPE

The AI-Based Automatic Examination Paper Evaluation System has significant potential for future enhancements and wider adoption in the education sector. Although the current system effectively evaluates descriptive answers using Machine Learning and Natural Language Processing, several improvements can be implemented to make the system more powerful, accurate, and scalable.

In the future, advanced deep learning models such as transformers and large language models can be integrated to better understand complex sentence structures and deeper semantic meanings. This would further improve evaluation accuracy, especially for long and concept-heavy answers. The system can also be extended to support multiple languages, enabling evaluation of descriptive answers written in regional and international languages.

Another important enhancement would be the inclusion of automated feedback generation. Instead of providing only marks, the system could give detailed feedback highlighting missing points, incorrect concepts, and areas for improvement. This would make the system more student-friendly and useful for learning purposes.

The system can be further expanded to support more subjects and academic domains by training models on diverse datasets. Integration with Learning Management Systems (LMS) and online examination platforms can allow seamless deployment in schools, colleges, and universities. Cloud deployment can also be introduced to handle large-scale examinations with a high number of users.

Additional security features such as role-based access control and secure data encryption can be implemented to improve data privacy and integrity. Performance optimization techniques can be applied to handle large volumes of answer scripts efficiently.

Overall, the future scope of this project is vast. With continuous improvement and integration of advanced AI technologies, the system can evolve into a complete intelligent evaluation platform that enhances transparency, efficiency, and quality in educational assessments.

10. REFERENCES

1. J. Wang and Y. Dong, ‘‘Measurement of text similarity: A survey,’’ Information, vol. 11, no. 9, p. 421, Aug. 2020.

2. M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, and C. Gao, ‘‘A survey on the techniques, applications, and performance of short text semantic similarity,’’ Concurrency Comput., Pract. Exper., vol. 33, no. 5, Mar. 2021.

3. M. S. M. Patil and M. S. Patil, ‘‘Evaluating Student descriptive answers using natural language processing,’’ Int. J. Eng. Res. Technol., vol. 3, no. 3, pp. 1716–1718, 2014.

4. P. Patil, S. Patil, V. Miniyar, and A. Bandal, ‘‘Subjective answer evaluation using machine learning,’’ Int. J. Pure Appl. Math., vol. 118, no. 24, pp. 1–13, 2018.

5. J. Muangprathub, S. Kajornkasirat, and A. Wanichsombat, ‘‘Document plagiarism detection using a new concept similarity in formal concept analysis,’’ J. Appl. Math., vol. 2021, pp. 1–10, Mar. 2021.

6. X. Hu and H. Xia, ‘‘Automated assessment system for subjective questions based on LSI,’’ in Proc. 3rd Int. Symp. Intell. Inf. Technol. Secur. Informat., Apr. 2010, pp. 250–254.

7. M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, ‘‘From word embeddings to document distances,’’ in Proc. Int. Conf. Mach. Learn., 2015, pp. 957–966.

8. C. Xia, T. He, W. Li, Z. Qin, and Z. Zou, ‘‘Similarity analysis of law documents based on Word2vec,’’ in Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C), Jul. 2019, pp. 354–357.

9. H. Mittal and M. S. Devi, ‘‘Subjective evaluation: A comparison of several statistical techniques,’’ Appl. Artif. Intell., vol. 32, no. 1, pp. 85–95, Jan. 2018.

10. L. A. Cutrone and M. Chang, ‘‘Automarking: Automatic assessment of open questions,’’ in Proc. 10th IEEE Int. Conf. Adv. Learn. Technol., Sousse, Tunisia, Jul. 2010, pp. 143–147.

11. G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, ‘‘A two-stage text feature selection algorithm for improving text classification,’’ Tech. Rep., 2021.

12. H. Mangassarian and H. Artail, ‘‘A general framework for subjective information extraction from unstructured English text,’’ Data Knowl. Eng., vol. 62, no. 2, pp. 352–367, Aug. 2007.

13. B. Oral, E. Emekligil, S. Arslan, and G. Eryigit, ‘‘Information extraction ˇ from text intensive and visually rich banking documents,’’ Inf. Process. Manage., vol. 57, no. 6, Nov. 2020, Art. no. 102361.

14. H. Khan, M. U. Asghar, M. Z. Asghar, G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, ‘‘Fake review classification using supervised machine learning,’’ in Proc. Pattern Recognit. Int. Workshops Challenges (ICPR). New York, NY, USA: Springer, 2021, pp. 269–288.

15. S. Afzal, M. Asim, A. R. Javed, M. O. Beg, and T. Baker, ‘‘URLdeepDetect: A deep learning approach for detecting malicious URLs using semantic vector models,’’ J. Netw. Syst. Manage., vol. 29, no. 3, pp. 1–27, Mar. 2021.

16. N. Madnani and A. Cahill, ‘‘Automated scoring: Beyond natural language processing,’’ in Proc.

27th Int. Conf. Comput. Linguistics (COLING), E. M. Bender, L. Derczynski, and P. Isabelle, Eds. Santa Fe, NM, USA: Association for Computational Linguistics, Aug. 2018, pp. 1099–1109.

17. Z. Lin, H. Wang, and S. I. McClean, ‘‘Measuring tree similarity for natural language processing based information retrieval,’’ in Proc. Int. Conf. Appl. Natural Lang. Inf. Syst. (NLDB) (Lecture Notes in Computer Science), vol. 6177, C. J. Hopfe, Y. Rezgui, E. Métais, A. D. Preece, and H. Li, Eds. Cardiff, U.K.: Springer, 2010, pp. 13–23.

18. G. Grefenstette, ‘‘Tokenization,’’ in Syntactic Wordclass Tagging. Springer, 1999, pp. 117–133.

19. K. Sirts and K. Peekman, ‘‘Evaluating sentence segmentation and word Tokenization systems on Estonian web texts,’’ in Proc. 9th Int. Conf. Baltic (HLT) (Frontiers in Artificial Intelligence and Applications) vol. 328, U. Andrius, V. Jurgita, K. Jolantai, and K. Danguole, Eds. Kaunas, Lithuania: IOS Press, Sep. 2020, pp. 174–181.

20. A. Schofield, M. Magnusson, and D. M. Mimno, ‘‘Pulling out the stops: Rethinking stopword removal for topic models,’’ in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics (EACL) vol. 2, M. Lapata, P. Blunsom, and A. Koller, Eds. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 432–436.

21. M. Çagatayli and E. Çelebi, ‘‘The effect of stemming and stop-wordremoval on automatic text classification in Turkish language,’’ in Proc. 22nd Int. Conf. Neural Inf. Process. (ICONIP) (Lecture Notes in Computer Science), vol. 9489, S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds. Istanbul, Turkey: Springer, 2015, pp. 168–176.

22. M. Divyapushpalakshmi and R. Ramalakshmi, ‘‘An efficient sentimental analysis using hybrid deep learning and optimization technique for Twitter using parts of speech (POS) tagging,’’ Int. J. Speech Technol., vol. 24, no. 2, pp. 329–339, Jun. 2021.

23. F. Camastra and G. Razi, ‘‘Italian text categorization with lemmatization and support vector machines,’’ in Neural Approaches to Dynamics of Signal Exchanges(Smart Innovation, Systems and Technologies), vol. 151, A. Esposito, M. Faúndez-Zanuy, F. C. Morabito, and E. Pasero, Eds. New York, NY, USA: Springer, 2020, pp. 47–54.

24. A. Jabbar, S. Iqbal, M. I. Tamimy, S. Hussain, and A. Akhunzada, ‘‘Empirical evaluation and study of text stemming algorithms,’’ Artif. Intell. Rev., vol. 53, no. 8, pp. 5559–5588, Dec. 2020.