Course: CS69099- Capstone Project

Product Name: AutoCorrect+ (AI-Enhanced Descriptive Answer Evaluation)

Proposal Prepared By:

Bhanu Siva Kumar Komanna - bkomman1@kent.edu - 811252220

Mani Surya Teja Kota - mkota1@kent.edu - 811262686

Veeraraghava Raju Lolabhattu - klolabha@kent.edu - 811301662

Gayathri Devi Atluri - gatluri@kent.edu - 811256395

Harika Malneedi - hmalneed@kent.edu -811255562

Sai Haritha Udatha - sudatha@kent.edu – 811238202

Date: 22 April 2024

Product Description

Evaluating descriptive answers is a time consuming task that requires a lot of manual work. It is prone to grader fatigue, bias, and the inconsistency that arises from varying standards. We have developed a solution to address these issues. Our automated grading system uses advanced NLP and ML techniques, enabling accurate and consistent evaluation of descriptive responses.

AutoCorrect+ uses several NLP tools, including NLTK for tokenization, stop word removal, and sklearn library for text similarity calculations using approaches such as cosine similarity and Jaccard distance. AutoCorrect+ uses these techniques to build a solid framework for efficiently grading descriptive answers. This system is also integrated with Django, which provides a user-friendly interface for both administrators and users. Educators may easily manage pending user registrations, check and analyze exam outcomes, add and remove courses and questions. Similarly, students can register, log in, attempt exams, view their results, and update their profiles seamlessly through the provided interfaces.

Overall, AutoCorrect+ goal is to improve the traditional manual grading process, bringing efficiency, accuracy, and scalability to educational assessment systems.

Product Value

AutoCorrect+, would be valuable for below reasons:

Time Efficiency: Manually evaluating descriptive answers is a time-consuming task, especially in large classes or online courses with many students. AutoCorrect+ automates this process, which reduces the workload of evaluators.

Consistency and fairness: We address the age-old issue of consistency and fairness in grading. Manual grading often leads to discrepancies and biases. However, AutoCorrect+ ensures a level playing field by providing fair and consistent evaluation of answers, regardless of who's grading them.

Scalability: As class sizes grow and assessments become more frequent, AutoCorrect+ scales effortlessly to meet the demand, without compromising on accuracy or quality.

Resource Optimization: Lastly, it saves resources. By reducing the need for extra grading materials, AutoCorrect+ helps schools save money and be more sustainable.

Building the Product

To create AutoCorrect+, we used the Natural Language Processing and Machine Learning libraries and frameworks. We begin by measuring semantic similarity between a student's answer and a pre-defined model answer. This involves a fusion of natural language processing techniques and machine learning algorithms. These are the steps involved:

Text Preprocessing: We start by preparing the textual data, tokenizing answers, removing stop words, and optionally performing stemming or lemmatization.

Text Vectorization: Next, textual data is converted into numerical vector representations using the CountVectorizer from scikit-learn, capturing word importance through frequencies.

Similarity Scoring: a. Cosine Similarity: Measures the angle between answer vectors, indicating text similarity. b. Jaccard Distance: Calculates distance between unique word sets, converted into a similarity score.

Answer Scoring: Scores from similarity measures are mapped to grading ranges and aggregated to derive a total score for the answer script.

User Interface Development: We've developed a user-friendly interface using the Django framework, empowering administrators to manage registrations, exams, and subjects. Students seamlessly register, attempt exams, and view results through the interface.

This methodology ensures efficient and accurate grading, making AutoCorrect+ a robust solution for automated evaluation

Determining Success

To assess the effectiveness of AutoCorrect+, we can use the following methods:

Benchmark Testing: We can create a representative collection of student responses and associated human-graded scores and compare it with autocorrect+ grades.

User Feedback: For realworld scenario, we can collect input from teachers, students, and other stakeholders who use AutoCorrect+ in real-world circumstances. Their feedback can provide vital insights on the system's usability, efficacy, and potential areas for development.

Deployment and Value Creation

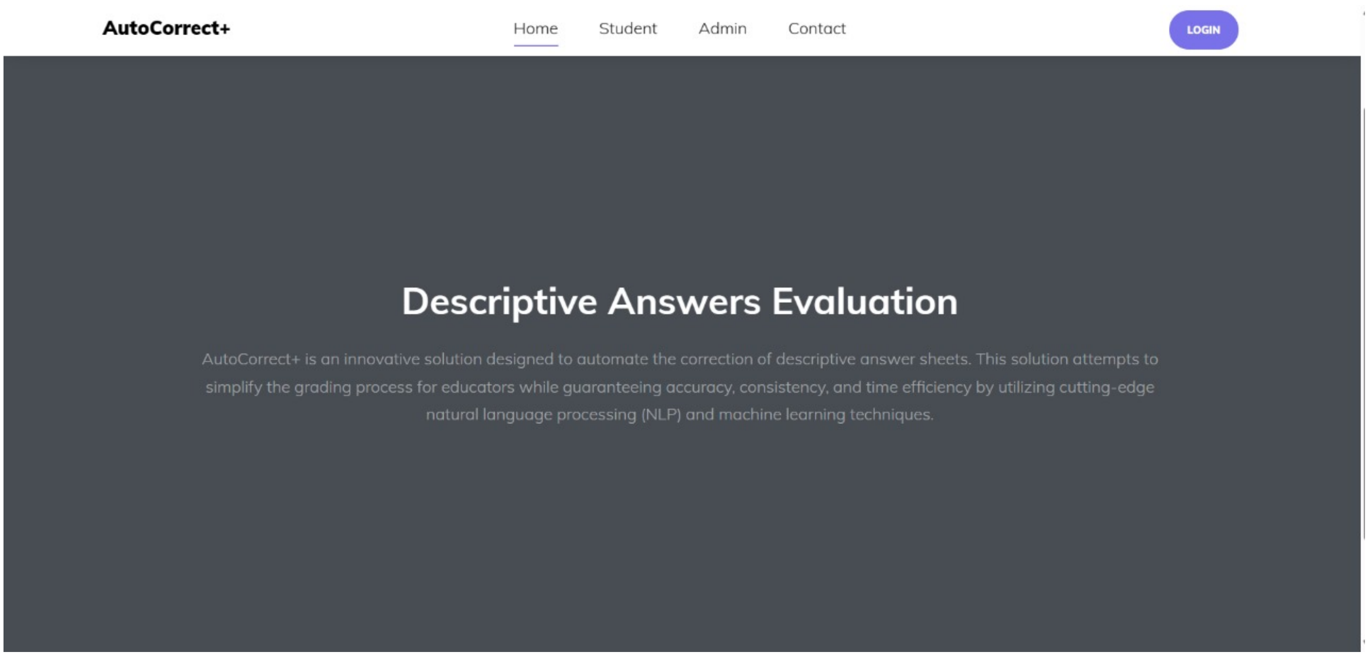
AutoCorrect+ can be used in a variety of educational environments to create value for different stakeholders:

Educational Institutions: AutoCorrect+ is compatible with learning management systems and evaluation platforms used by schools, colleges, and universities. This can help simplify the grading process, minimize instructors' workloads, and offer students with quick feedback.

Online Learning Platforms: As online education becomes more popular, AutoCorrect+ can be extremely useful for Online Courses and other e-learning platforms that require large-scale descriptive assignment grading.

Results

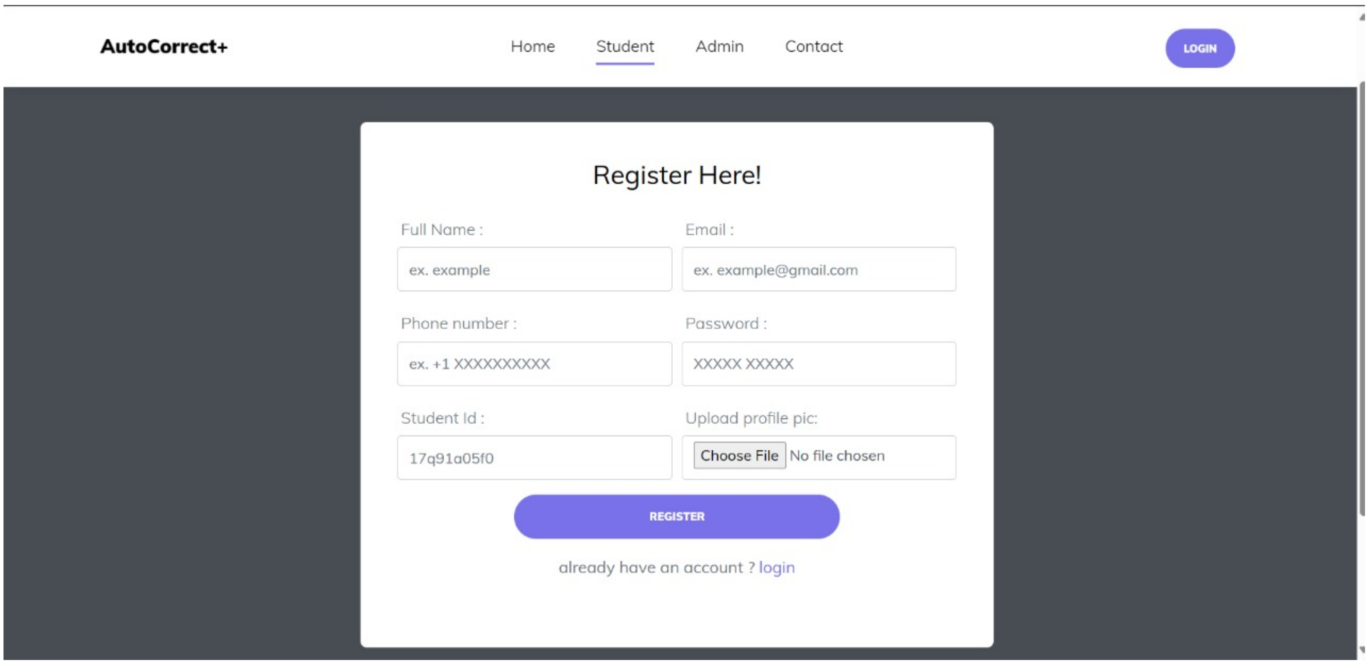
Login Page:



Descriptive Answers Evaluation

AutoCorrect+ is an innovative solution designed to automate the correction of descriptive answer sheets. This solution attempts to simplify the grading process for educators while guaranteeing accuracy, consistency, and time efficiency by utilizing cutting-edge natural language processing (NLP) and machine learning techniques.

User Registration:



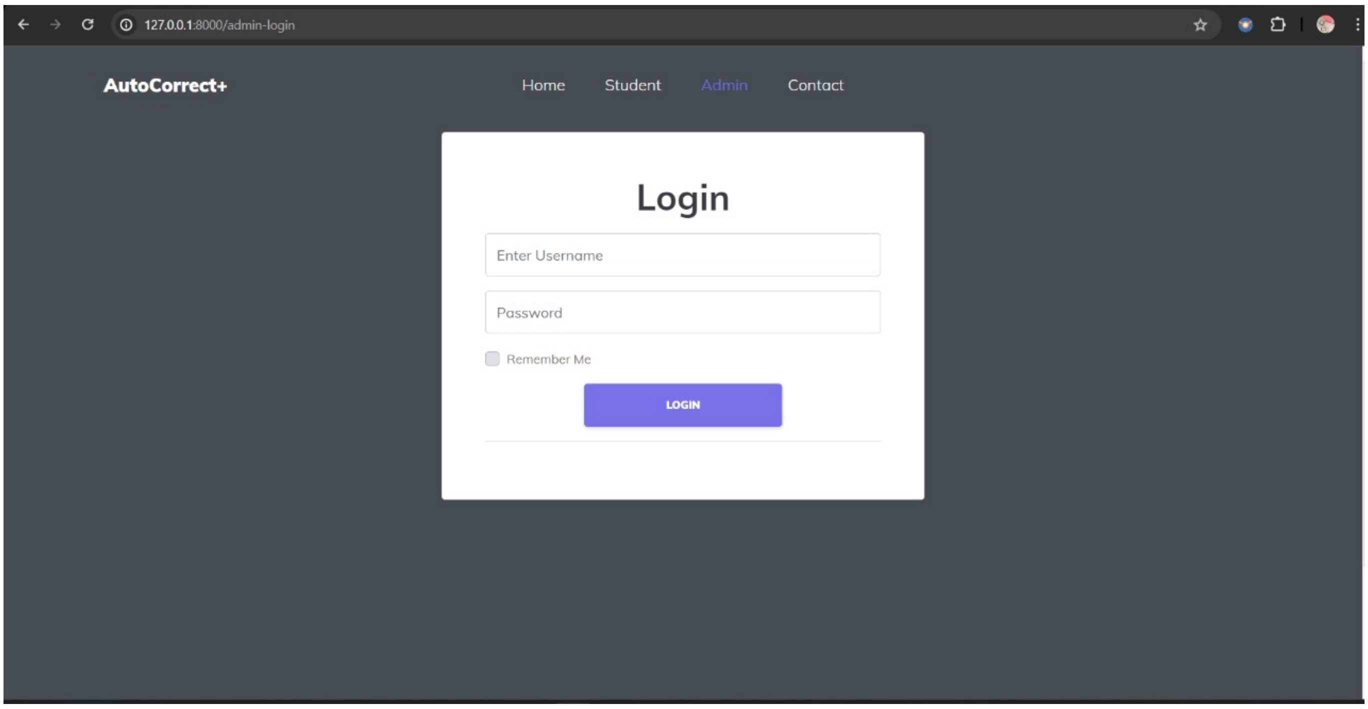
Register Here!

Full Name : ex. example	Email : ex. example@gmail.com
Phone number : ex. +1 XXXXXXXXXX	Password : XXXXX XXXXX
Student Id : 17q91a05f0	Upload profile pic: Choose File No file chosen

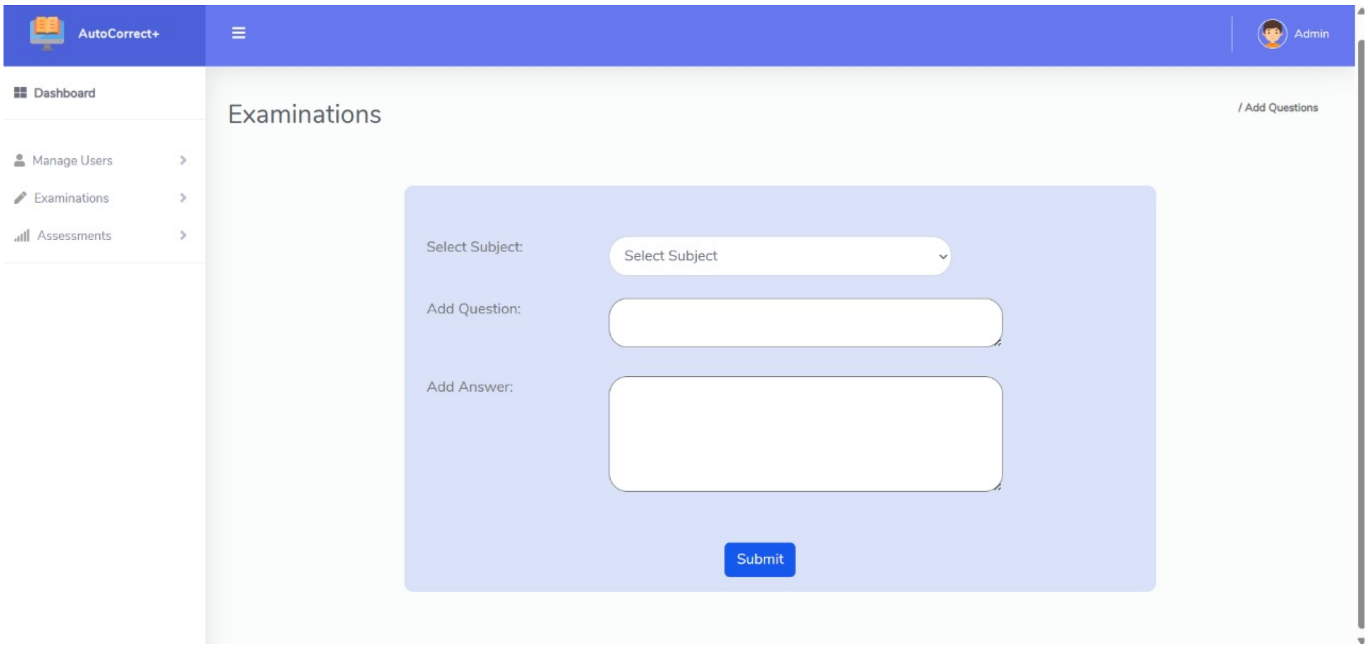
REGISTER

already have an account ? login

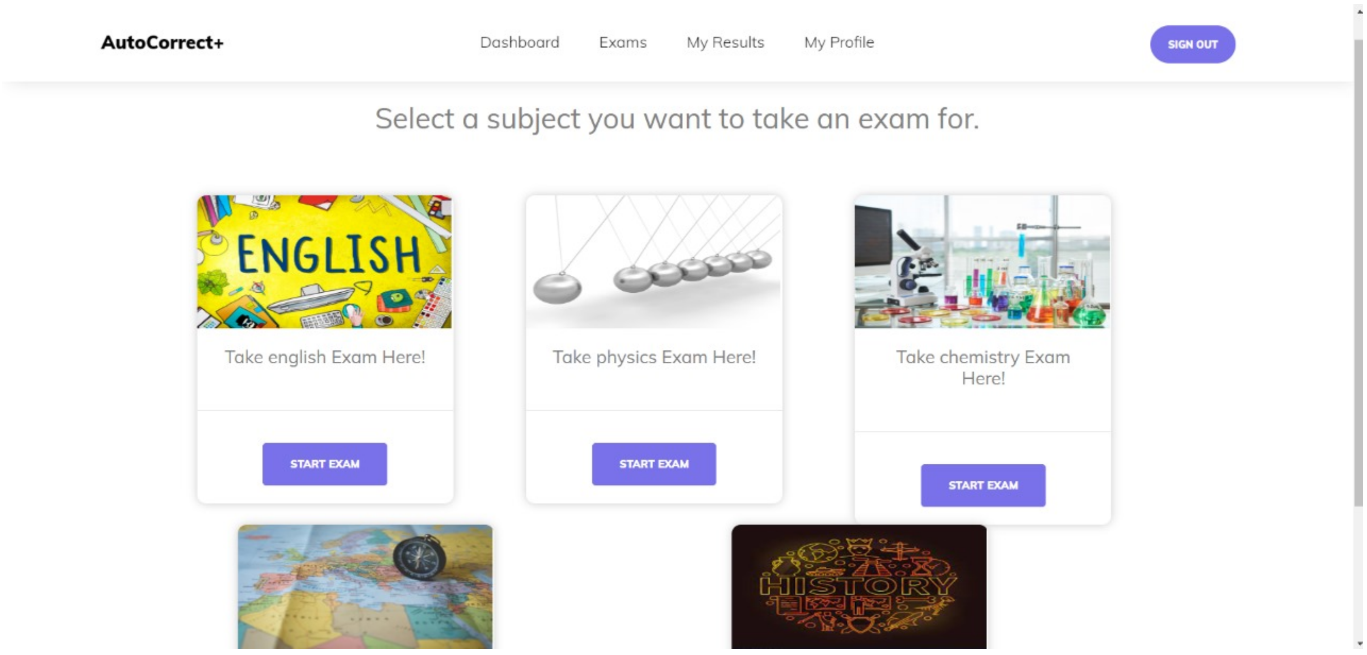
Admin Login:



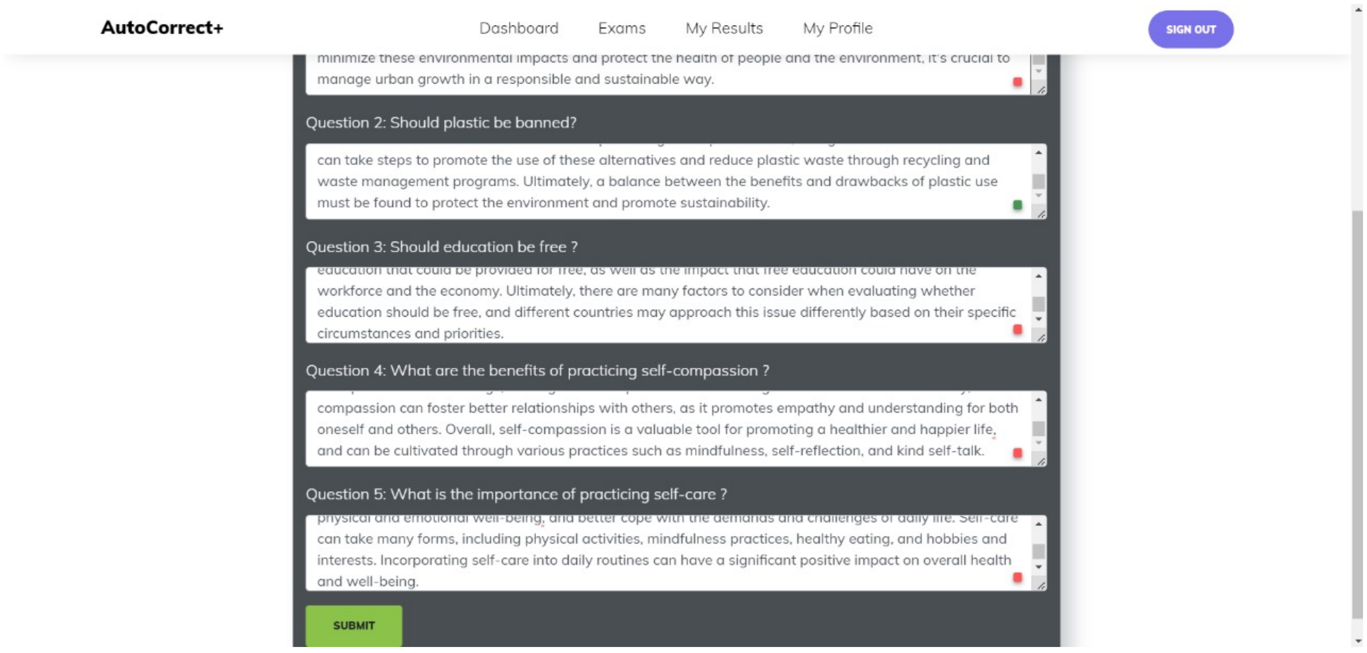
Admin can manage subjects, questions and answers



User Dashboard:



User answering questions:



minimize these environmental impacts and protect the health of people and the environment, it's crucial to manage urban growth in a responsible and sustainable way.

Question 2: Should plastic be banned?

can take steps to promote the use of these alternatives and reduce plastic waste through recycling and waste management programs. Ultimately, a balance between the benefits and drawbacks of plastic use must be found to protect the environment and promote sustainability.

Question 3: Should education be free ?

education that could be provided for free, as well as the impact that free education could have on the workforce and the economy. Ultimately, there are many factors to consider when evaluating whether education should be free, and different countries may approach this issue differently based on their specific circumstances and priorities.

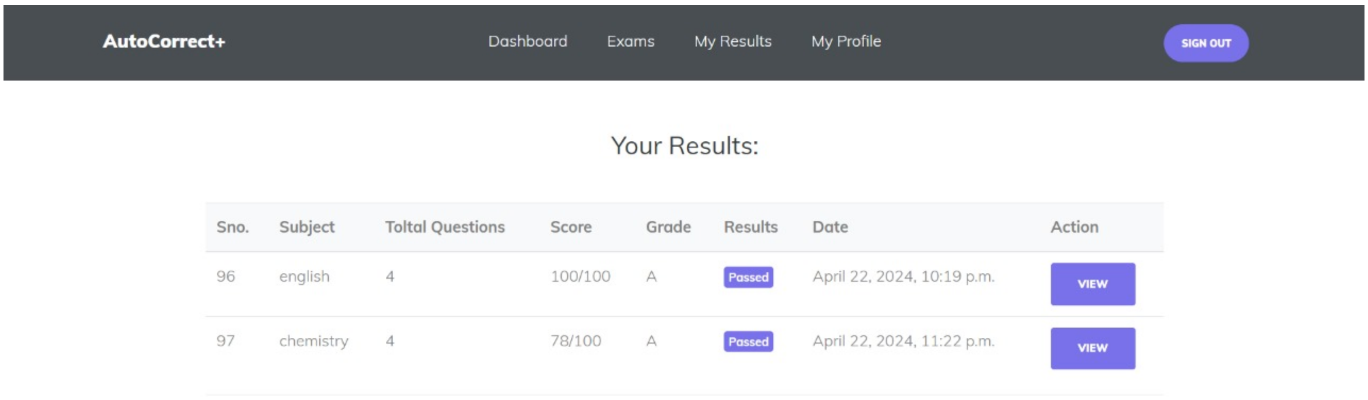
Question 4: What are the benefits of practicing self-compassion ?

compassion can foster better relationships with others, as it promotes empathy and understanding for both oneself and others. Overall, self-compassion is a valuable tool for promoting a healthier and happier life, and can be cultivated through various practices such as mindfulness, self-reflection, and kind self-talk.

Question 5: What is the importance of practicing self-care ?

physical and emotional well-being, and better cope with the demands and challenges of daily life. Self-care can take many forms, including physical activities, mindfulness practices, healthy eating, and hobbies and interests. Incorporating self-care into daily routines can have a significant positive impact on overall health and well-being.

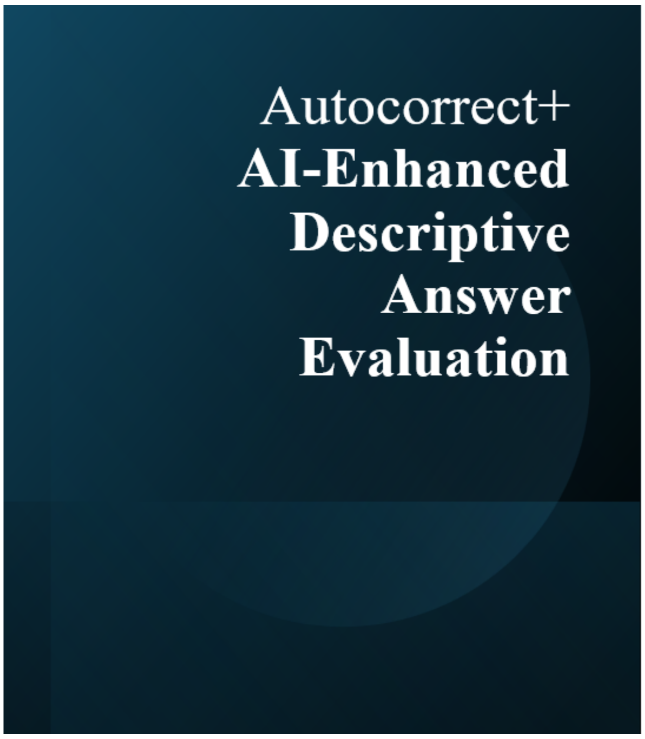
User Results:



Your Results:

Sno.	Subject	Toltal Questions	Score	Grade	Results	Date	Action
96	english	4	100/100	A	Passed	April 22, 2024, 10:19 p.m.	VIEW
97	chemistry	4	78/100	A	Passed	April 22, 2024, 11:22 p.m.	VIEW

Presentation



Autocorrect+
AI-Enhanced Descriptive Answer Evaluation

- Gayathri Devi Atluri- gatluri@kent.edu – 811256395
- Harika Malneedi- hmalneed@kent.edu- 811255562
- Sai Haritha Udatha- sudatha@kent.edu- 811238202
- Bhanu siva kumar komanna- bkomman1@kent.edu- 811252220
- Mani Surya Teja Kota- mkota1@kent.edu- 811262686
- Veeraraghava raju Lolabhattu- klolabha@kent.edu- 811301662

Abstract

AutoCorrect+ is an automated grading system that aims to accurately evaluate descriptive answers using natural language processing (NLP) and machine learning techniques. Traditional manual grading of subjective answers is time-consuming, prone to inconsistencies, and requires significant human effort. This project addresses those challenges by developing an AI-powered solution to automatically score descriptive answers.

Introduction

Problem Statement:

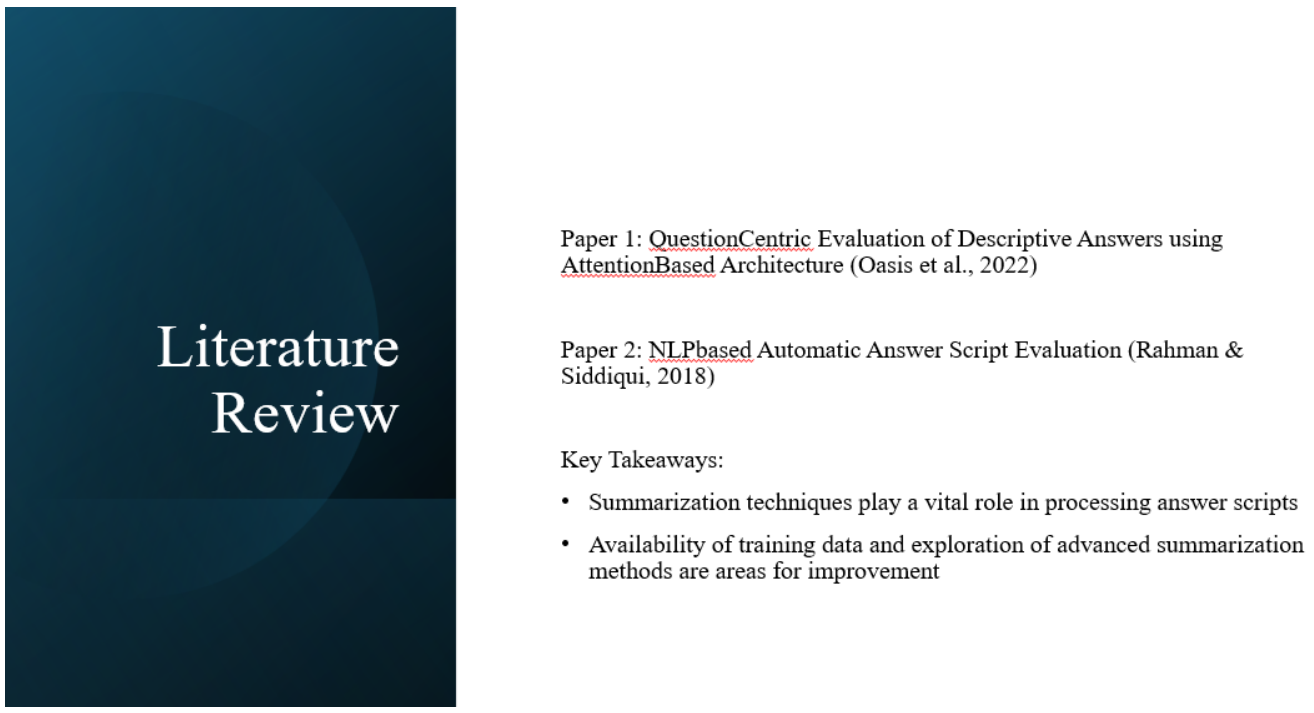
- Manual grading of descriptive answers is a laborious process
- Grader fatigue, bias, and varying standards lead to inconsistencies

Solution:

- An innovative solution to automate the grading of descriptive answers
- Utilizes advanced natural language processing (NLP) and machine learning (ML) techniques
- Enables accurate and consistent evaluation at scale

Core Technology:

- Measures semantic similarity between student's answer and model answer
- Text preprocessing, vectorization, and similarity scoring algorithms
- Techniques like cosine similarity, Jaccard distance, stop word removal



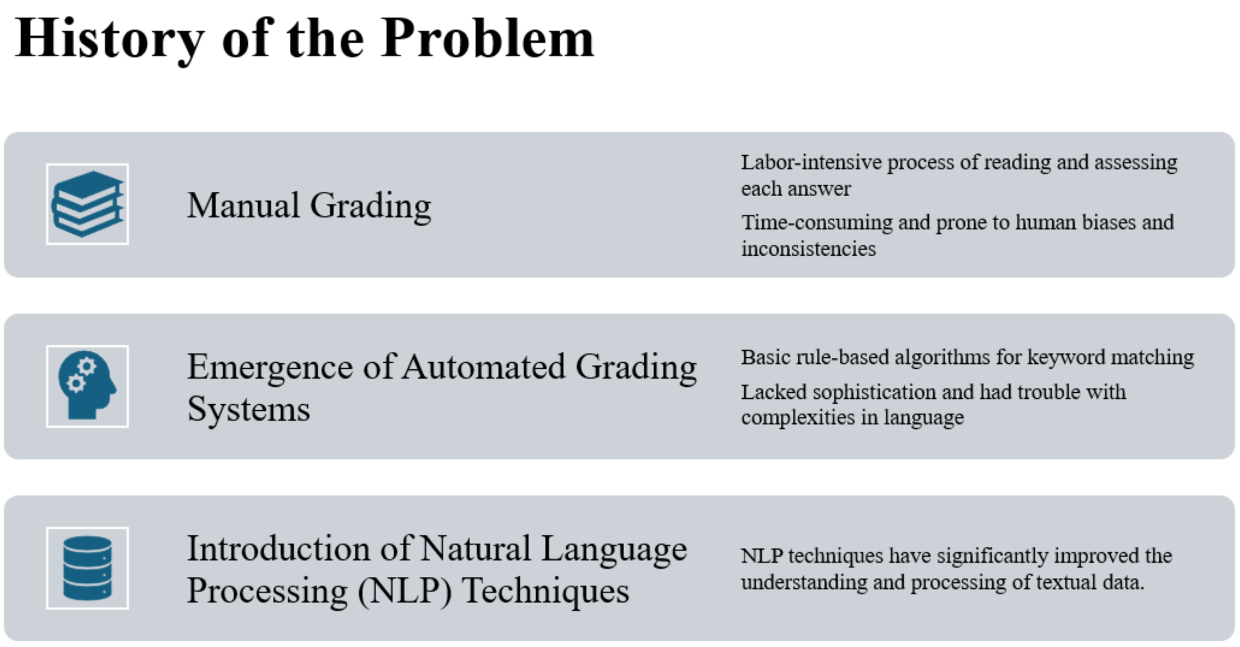
Literature Review

Paper 1: QuestionCentric Evaluation of Descriptive Answers using AttentionBased Architecture (Oasis et al., 2022)

Paper 2: NLPbased Automatic Answer Script Evaluation (Rahman & Siddiqui, 2018)

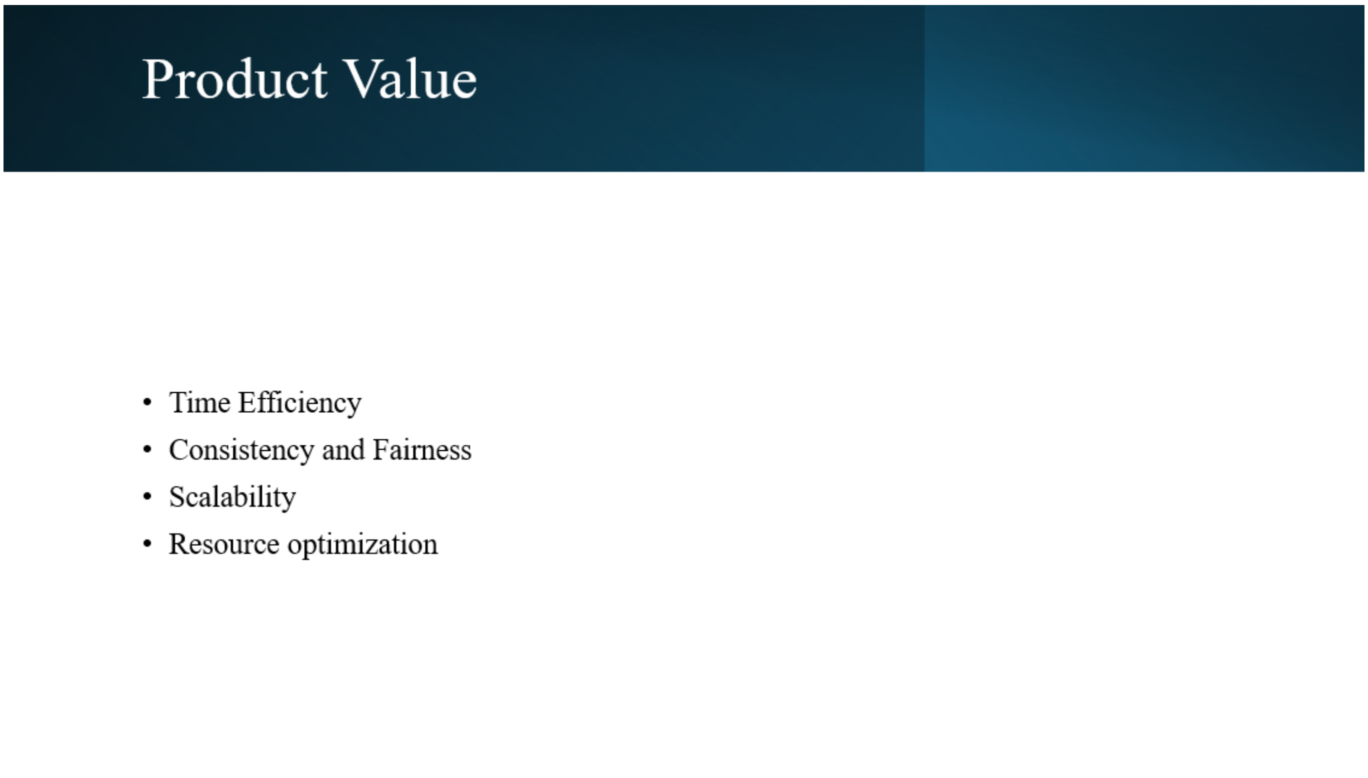
Key Takeaways:

- Summarization techniques play a vital role in processing answer scripts
- Availability of training data and exploration of advanced summarization methods are areas for improvement



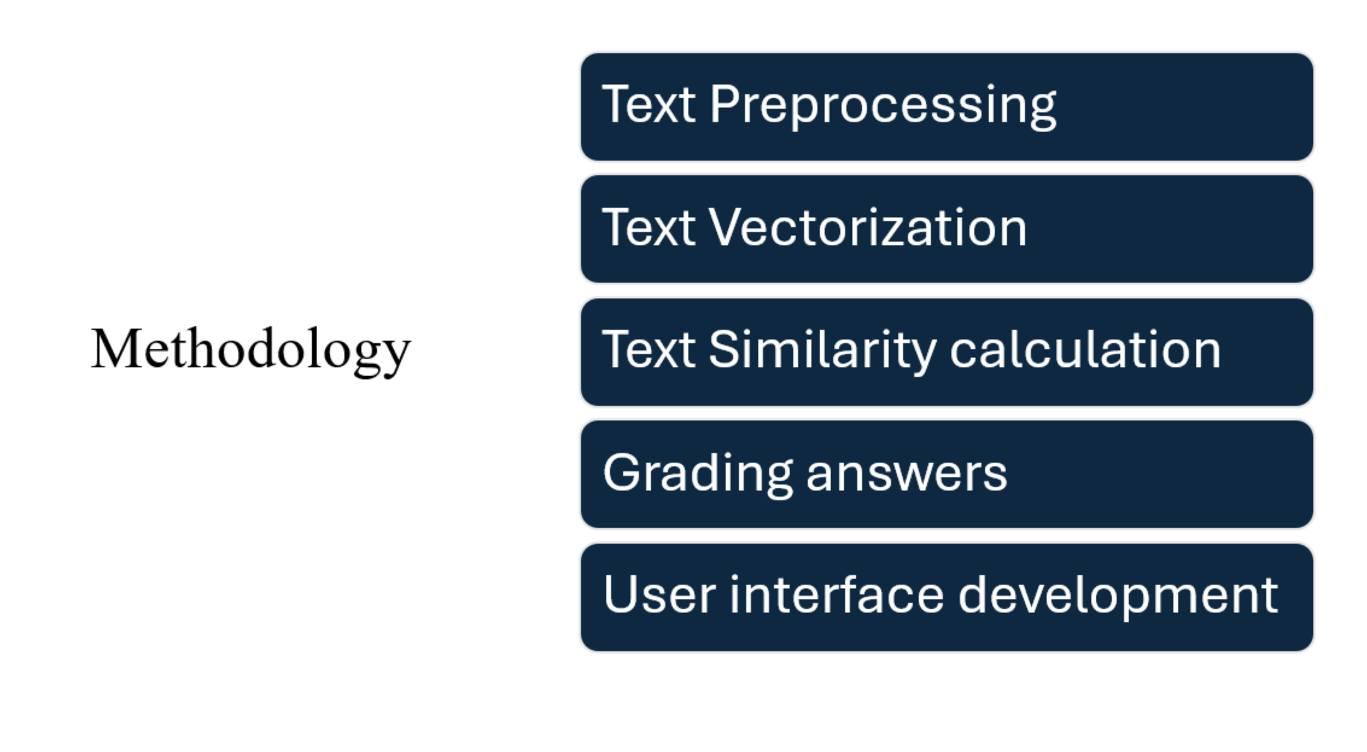
History of the Problem

Manual Grading	Labor-intensive process of reading and assessing each answer Time-consuming and prone to human biases and inconsistencies
Emergence of Automated Grading Systems	Basic rule-based algorithms for keyword matching Lacked sophistication and had trouble with complexities in language
Introduction of Natural Language Processing (NLP) Techniques	NLP techniques have significantly improved the understanding and processing of textual data.



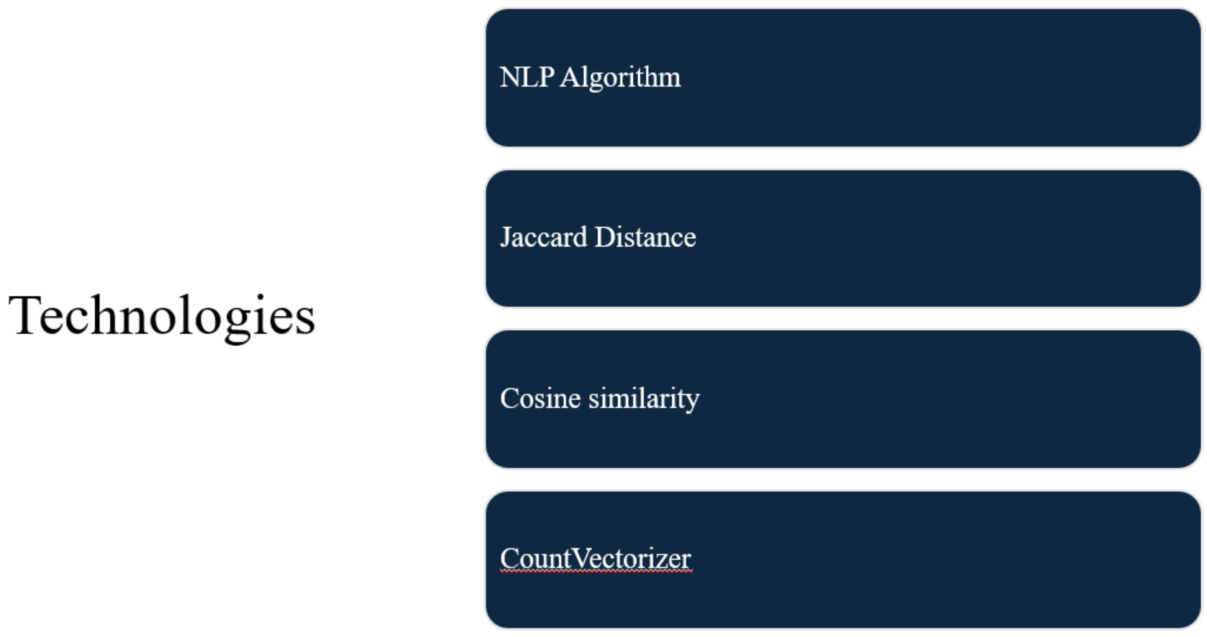
Product Value

- Time Efficiency
- Consistency and Fairness
- Scalability
- Resource optimization



Methodology

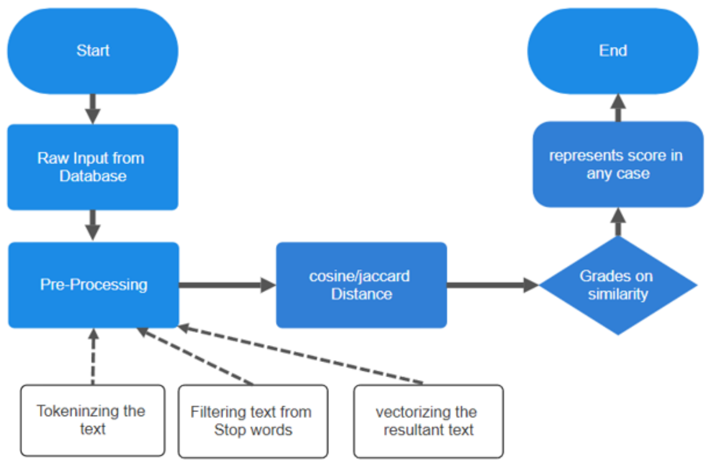
- Text Preprocessing
- Text Vectorization
- Text Similarity calculation
- Grading answers
- User interface development



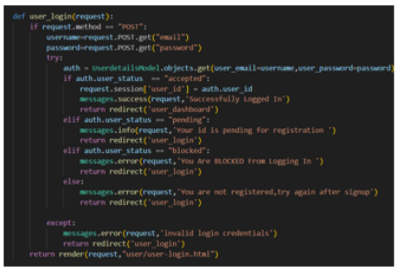
Technologies

- NLP Algorithm
- Jaccard Distance
- Cosine similarity
- CountVectorizer

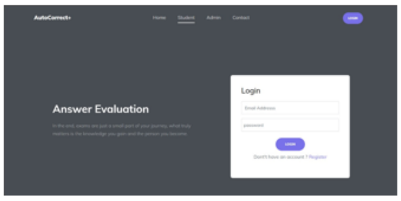
How the evaluation happens??



The Login Page



```python
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")
```



AutoCorrect+

Answer Evaluation

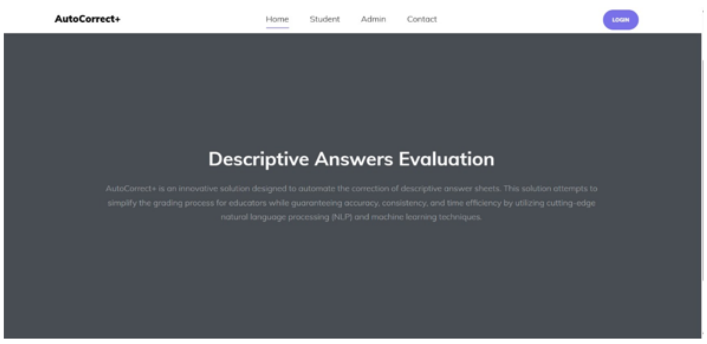
In the end, exams are just a small part of your journey, what truly matters is the knowledge you gain and the person you become.

Login

Email Address

password

Don't have an account ? Register

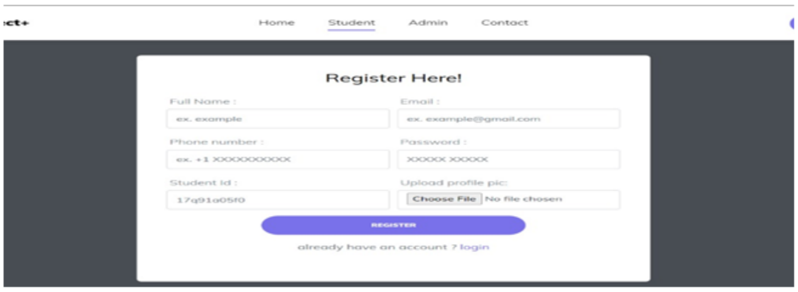


AutoCorrect+

Descriptive Answers Evaluation

AutoCorrect+ is an innovative solution designed to automate the correction of descriptive answer sheets. This solution attempts to simplify the grading process for educators while guaranteeing accuracy, consistency, and time efficiency by utilizing cutting-edge natural language processing (NLP) and machine learning techniques.

New User Registration



Register Here!

Full Name : ex. example

Email : ex. example@gmail.com

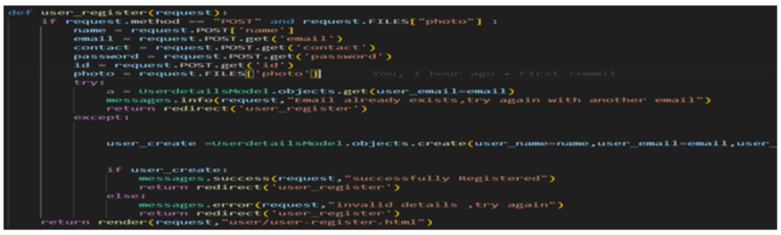
Phone number : ex. +1 XXXXXXXXXX

Password : XXXXXX XXXXXX

Student Id : 17q91a05f0

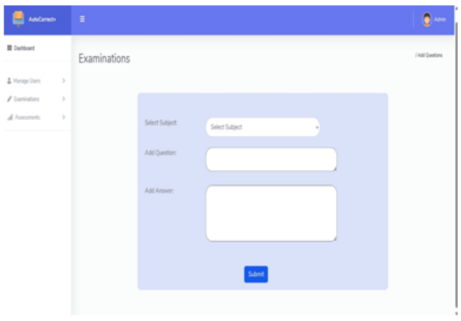
Upload profile pic: Choose File No file chosen

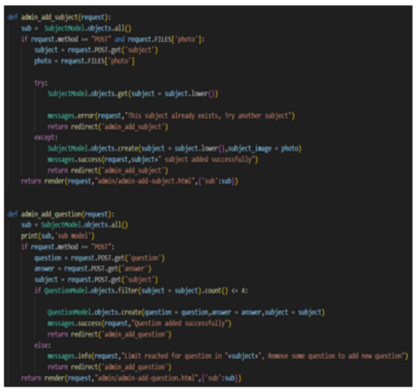
already have an account ? login



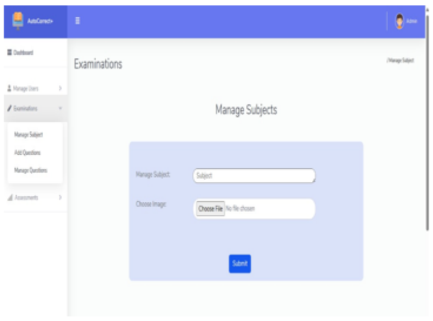
```python
def user_register(request):
    if request.method == "POST" and request.FILES["photo"] :
        name = request.POST['name']
        email = request.POST.get('email')
        contact = request.POST.get('contact')
        password = request.POST.get('password')
        id = request.POST.get('id')
        photo = request.FILES['photo']          You, 1 hour ago • First commit
        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_

            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")
```

Admin

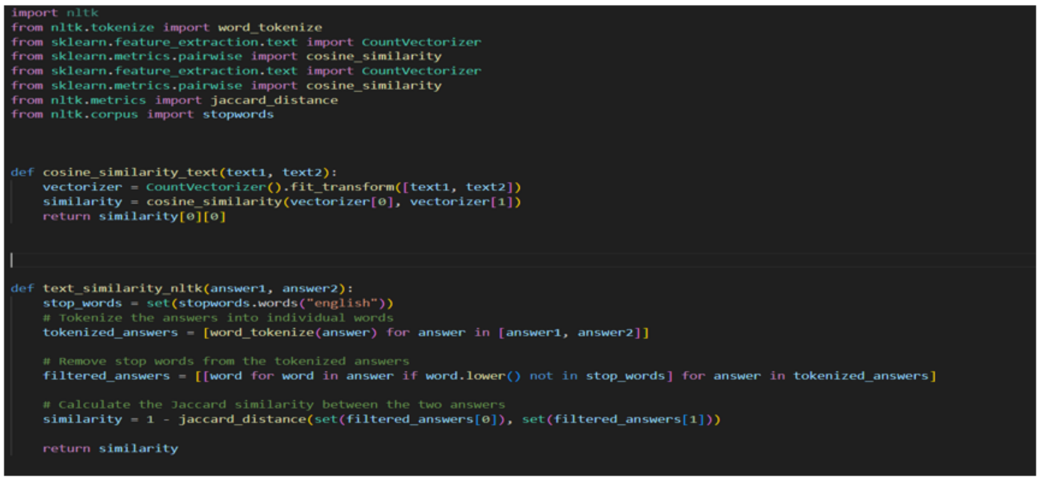




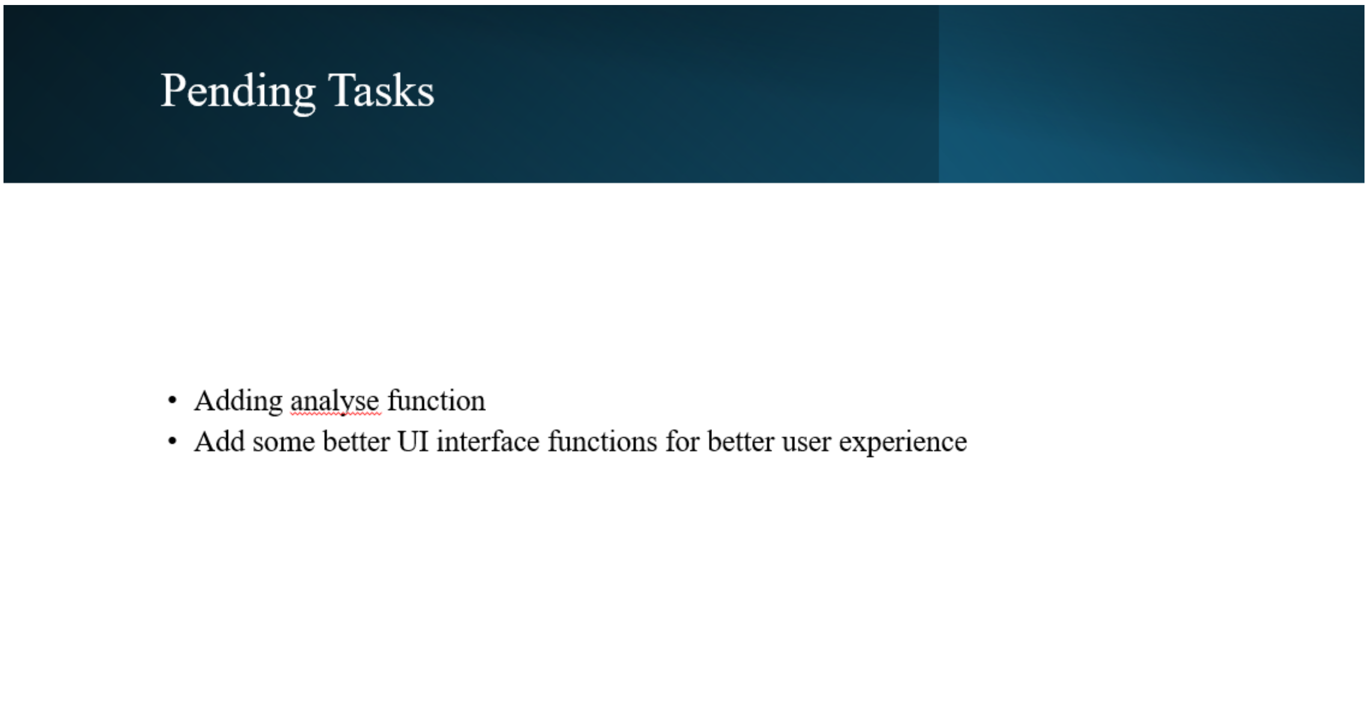
```python
def admin_add_subject(request):
    sub =  SubjectModel.objects.all()
    if request.method == "POST" and request.FILES['photo']:
        subject = request.POST.get('subject')
        photo = request.FILES['photo']

        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 question to add new question")
            return redirect('admin_add_question')
    return render(request,"admin/admin-add-question.html",{'sub':sub})
```



Text Similarity

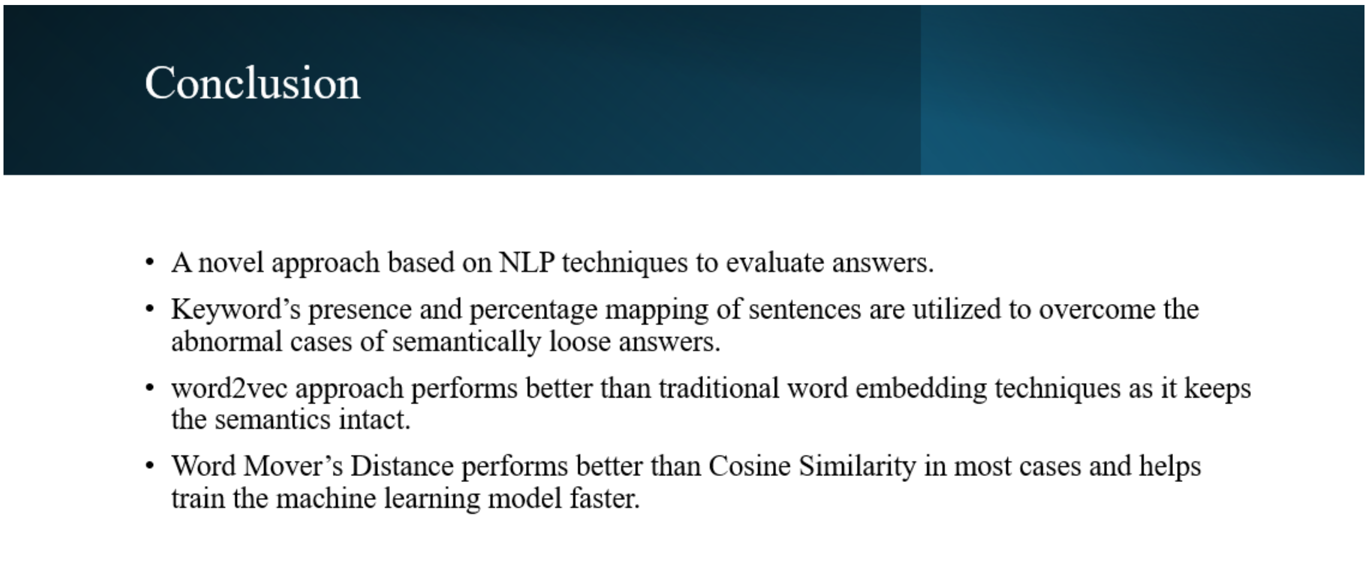


```python
import nltk
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from nltk.metrics import jaccard_distance
from nltk.corpus import stopwords


def cosine_similarity_text(text1, text2):
    vectorizer = CountVectorizer().fit_transform([text1, text2])
    similarity = cosine_similarity(vectorizer[0], vectorizer[1])
    return similarity[0][0]


def text_similarity_nltk(answer1, answer2):
    stop_words = set(stopwords.words("english"))
    # Tokenize the answers into individual words
    tokenized_answers = [word_tokenize(answer) for answer in [answer1, answer2]]

    # Remove stop words from the tokenized answers
    filtered_answers = [[word for word in answer if word.lower() not in stop_words] for answer in tokenized_answers]

    # Calculate the Jaccard similarity between the two answers
    similarity = 1 - jaccard_distance(set(filtered_answers[0]), set(filtered_answers[1]))

    return similarity
```



Pending Tasks

- Adding analyse function
- Add some better UI interface functions for better user experience



Conclusion

- A novel approach based on NLP techniques to evaluate answers.
- Keyword’s presence and percentage mapping of sentences are utilized to overcome the abnormal cases of semantically loose answers.
- word2vec approach performs better than traditional word embedding techniques as it keeps the semantics intact.
- Word Mover’s Distance performs better than Cosine Similarity in most cases and helps train the machine learning model faster.

Project Code

https://github.com/gayathrideviatluri/Capstone-Team1/

References

[1] Manjunath, Ravikumar & Kumar, S. & Guruswamy, Shiva. (2021). Automation of Answer Scripts Evaluation-A Review. 10.1007/978-981-33-6691-6_20.

[2] Mohammad Shaharyar Shaukat, Mohammed Tanzeem, Tameem Ahmad, Nesar Ahmad,Chapter 16 - Semantic similarity–based descriptive answer evaluation, Editor(s): Sarika Jain, Vishal Jain, Valentina Emilia

Balas,Web Semantics,Academic Press,2021,Pages 221-231,ISBN 9780128224687,https://doi.org/10.1016/B978-0-12-822468-7.00014-6.

[3] A. S. Oasis, A. E. M, D. Sharma, R. Sada and A. Arya, "Question-Centric Evaluation of Descriptive Answers using Attention-Based Architecture," 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2022, pp. 20-25, doi: 10.1109/Confluence52989.2022.9734117.

[4] Emlyn Hegarty-Kelly and Dr Aidan Mooney. 2021. Analysis of an automatic grading system within first year Computer Science programming modules. In Proceedings of the 5th Conference on Computing Education Practice (CEP '21). Association for Computing Machinery, New York, NY, USA, 17–20. https://doi.org/10.1145/3437914.3437973

[5] Rahman, Md & Siddiqui, Fazlul. (2018). NLP-based Automatic Answer Script Evaluation. 4. 35-42.