ST. TERESA’S COLLEGE

(AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

PROJECTREPORTON

PREDICTION ON IDENTIFYING SLOW LEARNERS

In partial fulfillment of the requirements for the

Award of the degree of

BSC COMPUTER APPLICATION [TRIPLE MAIN]

By

NANDANA SUNIL

III BSC COMPUTER APPLICATION [TRIPLE MAIN]

Register No: SB21CA018

Under the guidance of
Ms. REMYA CJ

Department of Computer

Applications2021-2024

THE FEAR OF THE LORD IS THE BEGINNING OF WISDOM

ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

CERTIFICATE

This is to certify that the project report on "**PREDICTION ON IDENTIFYING SLOW LEARNERS**" is a bona fide record of the work done by **NANDANA SUNIL (SB21CA018)** during the year2021-2024 and submitted in partial fulfillment of the requirement for the degree of **BSC COMPUTER APPLICATION [TRIPLE MAIN]** of Mahatma Gandhi University.

Submitted for end semester exam held on 22/03/ 2024

21/3/24

ST. TERESA'S COLLEGE
DEPT. OF COMP
ERNAKULAM

Head of the Department

21/3/24

Teacher in charge

22.3.24.

External Examiner

DECLARATION

I, **NANDANA SUNIL (Registerno:SB21CA018)**, **BSC Computer Application[Triple Main]**final year student of St. Teresa's College (Autonomous), Ernakulam, hereby declare that the project submitted named **PREDICTION ON IDENTIFYING SLOW LEARNERS** for the Bachelor's Degree in Computer Application [Triple Main] is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

Date: 21-03-2024

Nand

NANDANA SUNIL

Place: Ernakulam

ACKNOWLEDGEMENT

First and foremost, I would like to thank god almighty for the successful completion of my project. I express my sincere thanks to Provincial Manager Rev Dr. Sr Vinitha(CSST), Director Rev. Sr. Emeline (CSST) & Principal Dr. Alphonsa Vijaya Joseph of St. Teresa's college (AUTONOMOUS) for giving me an opportunity to undertake this project. I also extend my sincere gratitude to Ms. Remya CJ, my project guide for her constant support which helped in the successful completion of my project. I'm grateful to all the faculties of Department of Computer Applications for their valuable help and guidance during each stage of my project. I also extend my heartfelt thanks to Mr. Akash IT company, Ernakulam under whose guidance and support this project work has been carried out. Last but not the least, I would like to thank my parents and friends for motivating me and providing me the right environment for making this project work a great success.

NANDANA SUNIL

SYNOPSIS

Prediction on identifying slow learners is an Academic Project which gives students and teachers ideas on project ideas and research papers to be done or regarding the trend. Having a platform that give ideas of both under one roof is important, the time taken for discovering the project ideas is more than working on the same, so with the help of this prediction users can get a better understanding on ideas of their skills or knowledge, so they can standardize their time for working on it.

CONTENTS

1. INTRODUCTION

About Project

Prediction on identifying slow learner is a platform that helps in identifying those who have slow pace in acquiring new skill or knowledge. This application mainly focuses on primary school students. This project helps in identification of individual with slow learning.

About Organization

Founded in Cochin back in the year 1974, Riss project center started its journey as a premier Computer Education Center and later advanced into one of the most reputed brands for computer learning throughout India. After conducting extensive market research and analysis, Riss expert academicians and certified faculties collaborate with our leading computer technology company allies, such as, to develop pertinent certification courses.

Objectives of the Project and the Organization

The objective of the project is to systematically identify individuals with a slower learning pace, enabling targeted support and tailored education strategies. The project utilizes time and accuracy metrics in assessing responses to a given set of questions, providing insights into individual learning speeds and proficiency levels.

The main objective of the organization is to continuously optimize their customer's business through their world-class solutions, services and products. They ensure the success of the company by constantly and consistently satisfying the customers, shareholders, and employees.

2.SYSTEM ANALYSIS

Introduction

The primary goal of this project is to develop a predictive model that can identify students who might be categorized as slow learners based on various educational and behavioral features. A slow learner is a term used to describe an individual who take more time to grasp and learn information compared to their peers. It doesn't necessarily indicate a learning disability, but rather a slower pace in acquiring new skill or knowledge. This project helps in the identification of individuals with slower learning

Existing System

Natural Language Processing (NLP) tools, like NLTK, can be employed to analyze written or spoken language to gain insights into a student's learning patterns, comprehension, and expression. This may include sentiment analysis, topic modeling, or language complexity analysis

Proposed System

The proposed system integrates traditional educational assessments with advanced Natural Language Processing (NLP) techniques using NLTK. It collects diverse data including academic records and linguistic samples, preprocesses the data with NLTK, extracts relevant features, trains a predictive model, and evaluates its performance. Integrated into educational platforms, it ensures privacy and ethical compliance while continuously improving to provide educators with actionable insights for early intervention and personalized support for slow learners.

System Specification

The system collects diverse data, preprocesses it using NLTK for analysis, extracts relevant features, trains a predictive model, and evaluates its performance. Integrated into educational platforms, it ensures privacy, utilizes feedback loops for continuous improvement, and complies with ethical standards.

Operation System

The operating system requirements for the proposed system are versatile, as it can be deployed on various platforms. It should be compatible with major operating systems such as Windows, macOS, and Linux to accommodate different users' preferences. Additionally, it should be capable of running Python and supporting the NLTK library. Cloud-based solutions can also be considered for scalability and accessibility. Therefore, no specific operating system is mandated, but rather flexibility across different environments is essential for widespread adoption and usability.

Language or Software Packages

- **Python:** Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

- **MySQL**: MySQL is an open-source relational database management system. The SQL acronym stands for Structured Query Language. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment

- **NLP :**Natural Language Processing (NLP) is a branch of artificial intelligence (AI) focused on enabling computers to understand, interpret, and generate human language. It involves tasks like text classification, sentiment analysis, and language translation, using linguistic and statistical methods to analyze and process textual data. NLP has applications in various fields, including information retrieval, virtual assistants, and healthcare, revolutionizing human-computer interaction.

- **Html**: Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It defines the content and structure of web content.

Hardware and Software Requirements

Hardware requirement :

- Processor : Intel Core i5 or higher, AMD Ryzen 5 or higher
- RAM : 8 GB
- ROM : 10 GB
- Drive : SSD

Software requirement :

- Operating System : Windows, macOS or Linux
- Programming Language : Python
- Database Management System : MySQL
- Development tool :Html

3. SYSTEM DESIGN

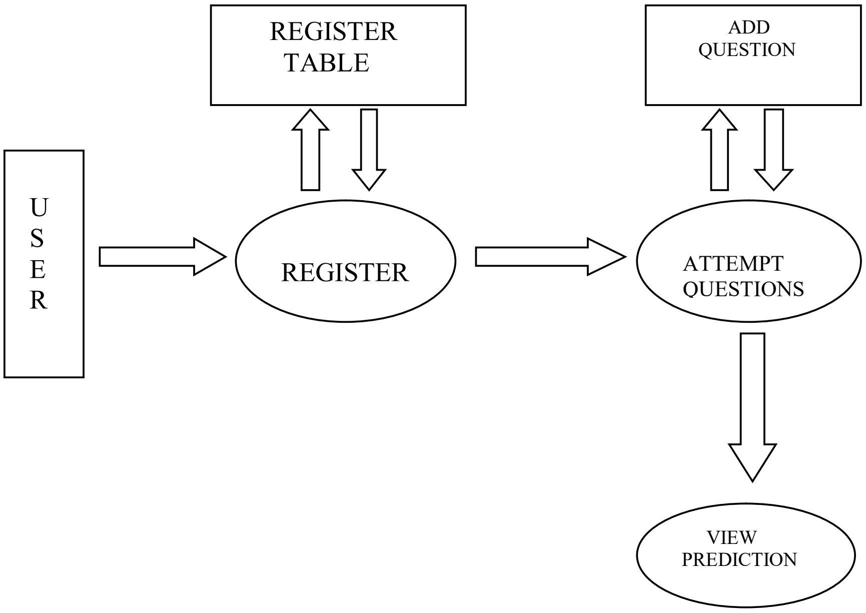
Introduction

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is a critical step in the software development lifecycle that bridges the gap between the requirements analysis and implementation phases

Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A DFD is often used as a primary step to create an overview of the system, which can later be elaborated. A DFD shows what will be the input of the system as well as the output. It clearly represents where the data will come from and go to and where the data will be stored.

DFD



Data Dictionary

A data dictionary is a collection of metadata that provides information about the data used in a database or information system. It serves as a reference guide for data elements and their definitions, as well as their relationships with other data elements in the system. A data dictionary is useful for ensuring data accuracy, consistency, and completeness. It can also help facilitate communication and collaboration among stakeholders who are involved in designing, developing, and maintaining a database or information system

Data Design

Database design is the organization of data according to a database model. A well designed database ensures data accuracy, consistency, and completeness and provides an intuitive and user-friendly interface for interacting with the data. Data design is the first design activity, which results in less complex, modular, and efficient program structure. The information domain model developed during analysis phase is transformed into data structures needed for implementing the software. The data objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in data dictionary provide a base for data design activity

4. SYSTEM DEVELOPMENT

Introduction

Software Development is the process of analyzing, designing, testing, implementation, and Software maintenance. It is called Software Development Life Cycle (SDLC). Different SDLC include waterfall, prototyping, iterative, incremental, spiral development, rapid application development and agile methodology

Process Description

Different processes of each module are given below:

- **Registration**

The user should enter their name, email, phone no, in there registration completed.

- **Taking Test**

The user is procedure to take the evaluation and given amount of question will appeared and the user should be answering those according to the knowledge.

- **Result**

The prediction is determined by the duration users take to complete the test. Longer completion times indicate a higher likelihood of being classified as a slow learner by the system. Therefore, the outcome is straightforward: users who take more time to finish the test are predicted to be slow learners

Code Design

```
from flask import Flask
from public import public
# from Student import Student
# from api import api
# from flask_cors import CORS

app=Flask(__name__)
# CORS(api)

app.secret_key='8555g'
app.register_blueprint(public)
# app.register_blueprint(Student,url_prefix='/Student')
```

```
app.run(debug=True,port=5338)

from flask import *
from database import*
from test import checkans

public=Blueprint('Public',__name__)

@public.route('/')
def index():
        return render_template('home.html')


@public.route('/Addquestions',methods=['get','post'])
def Addquestions():
        data={}
        if "submit" in request.form:
                q=request.form['question']
                a=request.form['answer']
                m=request.form['marks']
                t=request.form['time']
                ss="insert into questions values(null,'%s','%s','%s','%s')" %(q,a,m,t)
                insert(ss)
                return redirect(url_for('Public.Addquestions'))

  q="select * from questions"
        res=select(q)
        data['val']=res
        return render_template('Addquestions.html',data=data)


@public.route('/AttendExamss',methods=['get','post'])
def AttendExamss():
        data={}
        if "submit" in request.form:
                q=request.form['name']
                a=request.form['email']
                m=request.form['phone']
                sd="select * from participate where username='%s' OR email='%s' OR phone='%s'" %(q,a,m)
                print(sd)
                res2=select(sd)
                if res2:
                        pass
                else:
                        ss="insert into participate values(null,'%s','%s','%s')" %(q,a,m)
                        ii=insert(ss)
                        session['ii']=ii
                        return redirect(url_for('Public.ViewQuestions'))

return render_template('AttendExamss.html',data=data)

@public.route('/ViewQuestions',methods=['get','post'])
def ViewQuestions():
        data={}
```

Prediction on identifying slow learners

```
		q="select * from questions where question_id Not IN (select question_id from answers where 
	participate_id='%s')" %(session['ii'])
		res=select(q)
		if "submit" in request.form:
			qid=request.form['questionid']
			ans=request.form['answer']
			t=request.form['timer']

			q="select * from questions where question_id='%s' "%(qid)
			res=select(q)
			mark=res[0]['Maximum_mark']
			ti=res[0]['time']
			val=int(ti)-int(t)
			if val<0:
				out=1
			else:
				out=0

	# mark=checkans(res[0]['Maximum_mark'],maxmark)
			sim = checkans(res[0]['Description'], ans)
			print(type(sim))
			print(type(mark))
			omark = sim * float(mark)
			print("omark",omark)
			print("omaodivayoooooooooooooooooooooooooooooooooooooooooooooork",mark)

			q="insert into answers 
	values(null,'%s','%s','%s','%s','%s','%s')"%(qid,session['ii'],ans,omark,t,out)
			insert(q)
			q="select count(answer_id)as c from answers where participate_id='%s'" %(session['ii'])
			res11=select(q)
			if int(res11[0]['c'])>=5:
				return redirect(url_for('Public.index'))
			else:
				return redirect(url_for('Public.ViewQuestions'))


		data['ques']=res
		return render_template('ViewQuestions.html',data=data)

	  @public.route('/viewquestionbasedpredict')
	def viewquestionbasedpredict():
		data={}
		pid=request.args['pid']
		q="select * from answers inner join questions using(question_id) where participate_id='%s'" %(pid)
		res=select(q)
		data['val']=res
		qq="SELECT SUM(`Mark_awarded`) as ma,SUM(`out`) as o,count(answer_id) as aid FROM 
	answers WHERE participate_id='%s'" %(pid)
		res1=select(qq)
		data['ma']=res1[0]['ma']
		data['o']=res1[0]['o']

		ss=int(res1[0]['aid'])
		ma=int(res1[0]['ma'])
		o=int(res1[0]['o'])
```

Prediction on identifying slow learners

```
            kk=ma/ss
            print(kk)

            if kk<4:
                    os="failed"
            else:
                    os="passed"
            print(os)


        if o>3:
                    tt="More Time Taken"
            else:
                    tt="Less Time Taken"
            print(tt)

            if os=="failed" and tt=="More Time Taken":
                    dd="Slow Learning student"
            elif os=="failed" and tt=="Less Time Taken":
                    dd="Slow Learning student"
            elif os=="passed" and tt=="More Time Taken":
                    dd="Slow Learning student"
            elif os=="passed" and tt=="Less Time Taken":
                    dd="Fast Learning student"
            print(dd)
            data['os']=os
            data['tt']=tt
            data['dd']=dd

            return render_template('viewquestionbasedpredict.html',data=data)

    @public.route('/viewparticipant')
    def viewparticipant():
            data={}
            q="select * from participate"
            res=select(q)
            data['p']=res

    return render_template('viewparticipant.html',data=data)



    from flask import *
    from database import*
    import uuid
    from test import checkans



    Student=Blueprint('Student',__name__)


    @Student.route("/Studenthome")
    def Studenthome():
```

```
    return render_template('Studenthome.html')


@Student.route("/viewsubjects",methods=['post','get'])
def viewsubjects():
    data={}
    he="SELECT * FROM subjects INNER JOIN `courses` ON
`courses`.`Couse_id`=`subjects`.`Course_id` where `courses`.`Couse_id`=(SELECT `Course_id`FROM
`students`WHERE Student_id='%s')"%(session['s_id'])
    data['vsub']=select(he)
    return render_template('viewsubjects.html',data=data)

@Student.route("/viewteacher",methods=['post','get'])
def viewteacher():
    data={}
    vt="select * from teachers"
    data['teacherv']=select(vt)

    return render_template('viewteacher.html',data=data)

@Student.route("/Studentviewexam",methods=['post','get'])
def Studentviewexam():
    data={}
    sve="SELECT * FROM `exams` INNER JOIN `subjects` USING(`Subject_id`) where
Course_id=(SELECT `Course_id`FROM `students`WHERE `Student_id`='%s')"%(session['s_id'])
    res=select(sve)
    data['stve']=res
    return render_template('Studentviewexam.html',data=data)


@Student.route("/Attendexam")
def Attendexam():
    data={}
    att="SELECT * FROM `exams` INNER JOIN `subjects` USING(`Subject_id`) where
Course_id=(SELECT `Course_id`FROM `students`WHERE `Student_id`='%s')"%(session['s_id'])
    data['view']=select(att)


    return render_template('Attendexam.html',data=data)

@Student.route("/Attend",methods=['post','get'])
def Attend():
    data={}
    qid=request.args['q_id']


    if 'action' in request.args:
        action=request.args['action']
        qid=request.args['q_id']

    else:
        action=None
    if 'sub' in request.form:
        ad=request.form['Answer_details']
        ins="insert into answers values(null,'%s','%s','%s','pending')"%(qid,session['s_id'],ad)
        insert(ins)
```

```
		return render_template('Attend.html',data=data)


	@Student.route("/Studentviewresult")
	def Studentviewresult():
		data={}
		exam=request.args['eid']
		q="select * from result inner join exams using(Exam_id) inner join subjects using(Subject_id) where 
	Exam_id='%s'"%(exam)
		data['view']=select(q)
		return render_template('Studentviewresult.html',data=data)


	@Student.route("/payment",methods=['post','get'])
	def payment():
		data={}
		eid=request.args['eid']
		amount=request.args['Amount']
		data['amount']=amount
		dd="select * from request where Student_id='%s' and Exam_id='%s'"%(session['s_id'],eid)
		res=select(dd)
		if res:
			flash("Already paid.......!")
			return redirect(url_for('Student.Studentviewexam'))
		if 'sub' in request.form:
			ins="insert into request values(null,'%s','%s',curdate(),'paid')"%(session['s_id'],eid)
			insert(ins)
			se="insert into payment 
	values(null,'%s','%s','%s',curdate())"%(eid,session['s_id'],data['amount'])
			insert(se)
			flash("paid...........!")
			return redirect(url_for('Student.Studentviewexam'))
		return render_template('payment.html',data=data)


	@Student.route('/view_questions')
	def view_questions():
		data={}
		eid=request.args['eid']
		bn="SELECT * FROM `request`WHERE `Exam_id`='%s' and 
	Student_id='%s'"%(eid,session['s_id'])
		fd=select(bn)
		if fd:
			q="select * from questions where Exam_id='%s'"%(eid)
			data['view']=select(q)
		else:
			flash("Pay Exam fee..........!")
			return redirect(url_for('Student.Studentviewexam'))
		return render_template('view_questions.html',data=data,eid=eid)

	@Student.route('/answers',methods=['post','get'])
	def answers():
		qid=request.args['qid']
		eid=request.args['eid']
		maxmark=request.args['maxmark']
```

Prediction on identifying slow learners

```
    ss="select * from answers where question_id='%s' and Student_id='%s'"%(qid,session['s_id'])
    sd=select(ss)
    if sd:
        flash("Already attended..!")
        return redirect(url_for('Student.Attendexam'))
    if 'submit' in request.form:
        ans=request.form['ans']
        q="select * from questions where question_id='%s' "%(qid)
        res=select(q)
        mark=10
        # mark=checkans(res[0]['Maximum_mark'],maxmark)
        sim = checkans(res[0]['Description'], ans)
        omark = sim * mark
        print("omark",omark)
        print("omaodivayoooooooooooooooooooooooooooooooooooooooooork",mark)

        q="insert into answers values(null,'%s','%s','%s','%s')"%(qid,session['s_id'],ans,omark)
        insert(q)
        h="select * from participation where Exam_id='%s' and 
Student_id='%s'"%(eid,session['s_id'])
        dd=select(h)
        if dd:
            pass
        else:
            k="insert into participation values(null,'%s','%s')"%(eid,session['s_id'])
            insert(k)
        return redirect(url_for('Student.Studentviewexam'))
    return render_template('answers.html')
```

```
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
import re, math
from collections import Counter

ps=PorterStemmer()

WORD = re.compile(r'\w+')
def checkans(ans,oans):
    def sr(txt):
        stop_words=set(stopwords.words('english'))
        word_tokens=word_tokenize(txt)
        filtered_sentence=[w for w in word_tokens if not w.lower() in stop_words]
        filtered_sentence1=[]
        for i in filtered_sentence:
            #filtered_sentence1.append(ps.stem(i))
            filtered_sentence1.append(i)
        print("*************",filtered_sentence1)

        return ' '.join(filtered_sentence1)
    def get_cosine(vec1, vec2):
        intersection = set(vec1.keys()) & set(vec2.keys())
        numerator = sum([vec1[x] * vec2[x] for x in intersection])
        sum1 = sum([vec1[x]**2 for x in vec1.keys()])
        sum2 = sum([vec2[x]**2 for x in vec2.keys()])
        denominator = math.sqrt(sum1) * math.sqrt(sum2)

        if not denominator:
          return 0.0
        else:
          return float(numerator) / denominator

    def text_to_vector(text):
        words = WORD.findall(text)
        return Counter(words)


    text1=sr(ans.lower())
    text2=sr(oans.lower())
    print(text1)
    print(text2)

    vector1=text_to_vector(str(text1))
    vector2=text_to_vector(text2)

    cosine=get_cosine(vector1,vector2)
    print ('Cosine:', cosine)
    #      #omark = cosine * 10
    # return cosine
    omark = cosine * 10
    print(omark,"huhuhuhuhu")
    return om
```

5.SYSTEM TESTING
AND IMPLEMENTATION

Introduction

Software testing is a crucial process in the software development life cycle that involves evaluating the quality and functionality of software applications. The main goal of software testing is to identify defects or errors in the application before it is released to the end-users. The software testing process typically involves several steps, including test planning, test design, test execution, and test reporting.

System Implementation

Implementation is the action that must follow any preliminary thinking for something to happen. Software/hardware implementations should always be designed with the end user in mind and the implementation process usually benefits from user involvement and support from managers and other top executives in the company. If users participate in the design and implementation of the system, ideally it will serve their business objectives more accurately and reflect their priorities and the ways in which they prefer to work

Debugging

Debugging is the process of identifying and resolving issues or errors in software applications. It is a critical step in the software development life cycle and is aimed at improving the quality and functionality of the application. During the development process, errors or bugs can occur in the code that can cause the application to behave in unexpected ways or not work at all. Debugging involves identifying and diagnosing these errors, tracing their root cause, and then making the necessary changes to fix them.

Different types of debugging methods used in this system are:

- **Unit Testing:** The application was divided into smaller components and tested individually. Each code was executed separately to ensure accuracy

- **Integration Testing:** Each small component was integrated or combined into a module to ensure that each module works properly when put together. This was done to check connectivity between module

- **System Testing:** The system was tested by combining every module. This was to ensure that each process have a particular order. This was to ensure that the system does not crash while using

Scope for future enhancement

In future this interface can be further developed into an application or a website

6.CONCLUSION

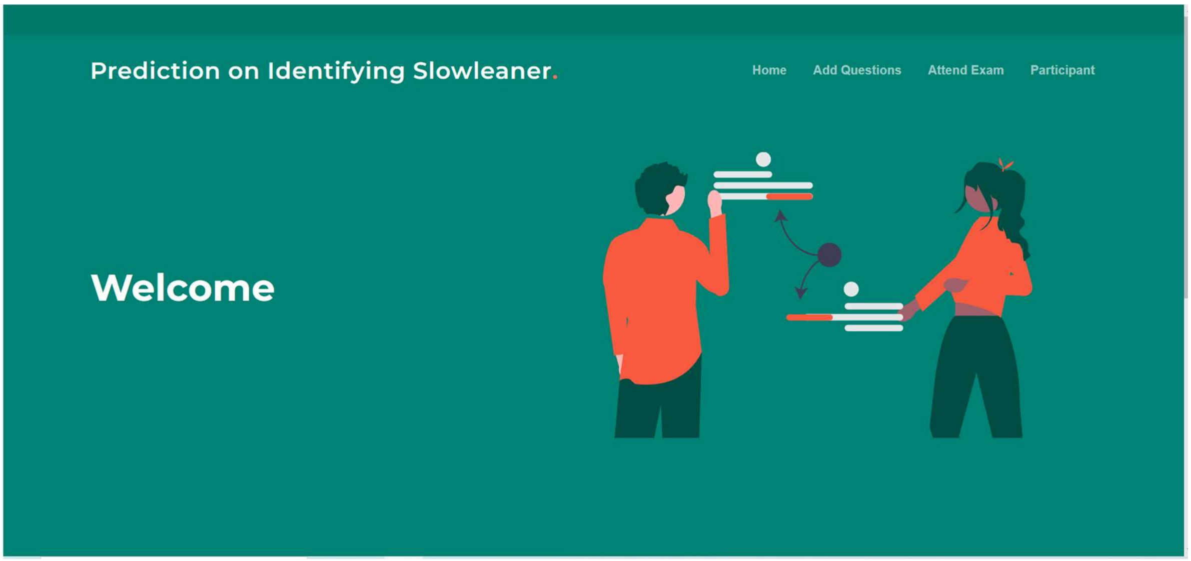
6.Conclusion:

In conclusion, this project on identifying slow learners through the analysis of time and accuracy in question responses holds the potential to revolutionize education by fostering tailored learning approaches. The insights gained can pave the way for more inclusive and effective educational strategies, ensuring that each learner receives the support necessary for their unique pace and proficiency.

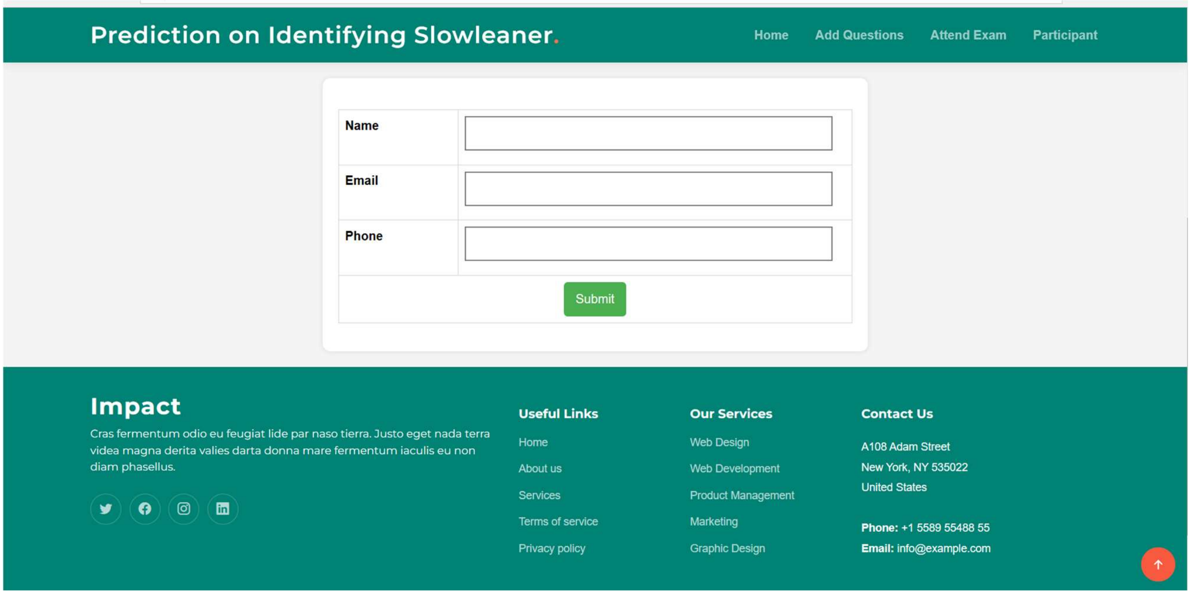
.

7.APPENDIX

Home Page



Registration Page



Name	
Email	
Phone	

Impact

Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra videa magna derita valies darta donna mare fermentum iaculis eu non diam phasellus.

Useful Links

Home
About us
Services
Terms of service
Privacy policy

Our Services

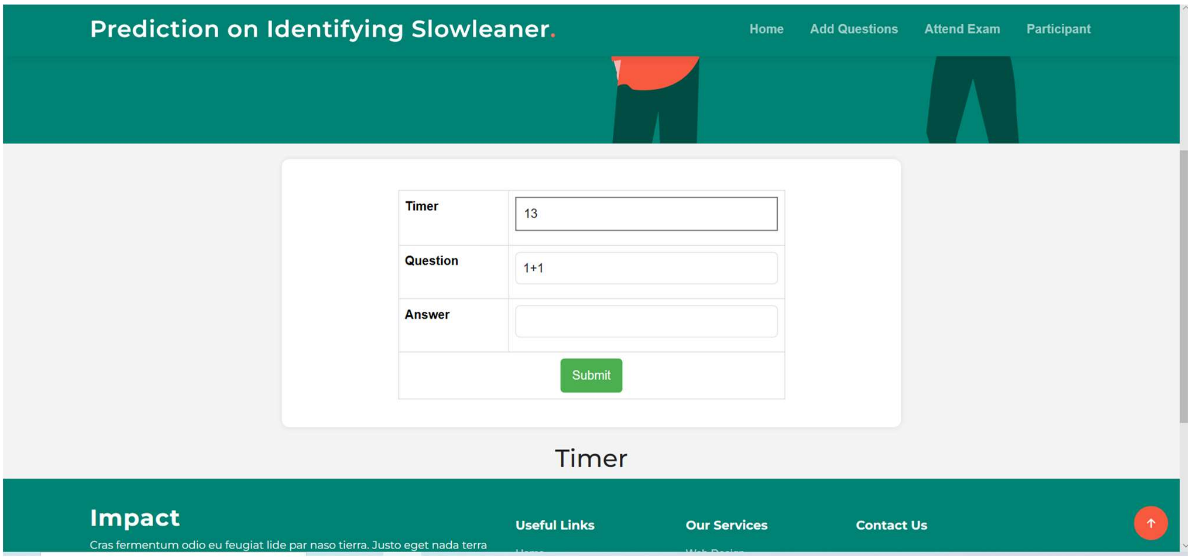
Web Design
Web Development
Product Management
Marketing
Graphic Design

Contact Us

A108 Adam Street
New York, NY 535022
United States

Phone: +1 5589 55488 55
Email: info@example.com

Attempt Questions

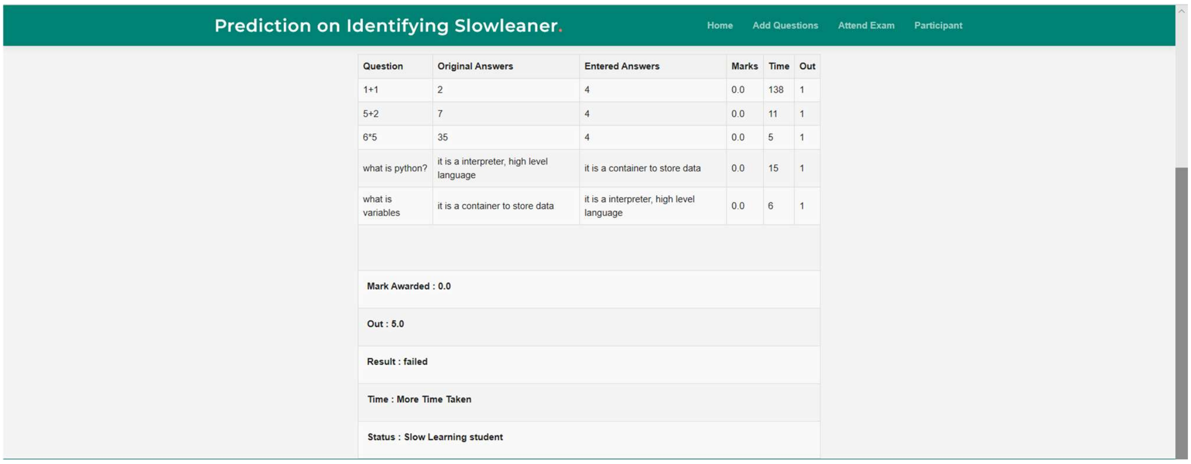


Timer	13
Question	1+1
Answer	

Timer

Cras fermentum odio eu feugiat lide par naso tierra. Justo eget nada terra

Result



Question	Original Answers	Entered Answers	Marks	Time	Out
1+1	2	4	0.0	138	1
5+2	7	4	0.0	11	1
6*5	35	4	0.0	5	1
what is python?	it is a interpreter, high level language	it is a container to store data	0.0	15	1
what is variables	it is a container to store data	it is a interpreter, high level language	0.0	6	1

Mark Awarded : 0.0

Out : 5.0

Result : failed

Time : More Time Taken

Status : Slow Learning student

8.BIBLIOGRAPHY

References

- Beena, Joseph, and Sajimon Abraham. "Identifying slow learners in an e-learning environment using k-means clustering approach." *Knowledge Management & E-Learning* 15.4 (2023):539.

- Murdiyanto, Tri, Dwi Antari Wijayanti, and Anny Sovia. "Identify Slow Learners in Math: Case Study in Rural Schools." International Journal of Interactive Mobile Technologies 17.6 (2023)

- Malik, Najma Iqbal, Ghazala Rehman, and Rubina Hanif. "Effect of Academic Interventions on the Developmental Skills of Slow Learners." Pakistan Journal of Psychological Research 27.1 (2012).

Prediction on identifying slow learners