A

Major Project

On

PERFORMANCE ANALYSIS ON STUDENT FEEDBACK USING MACHINE LEARNING ALGORITHMS

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

RASNOLA GAYATHRI (207R1A05N6)

VANAPARTHI ARCHANA (207R1A05P4)

KUKKAMALA DEEPESH (217R5A0524)

Under the Guidance of

Dr.G.MADHUKAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2020-2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"PERFORMANCE ANALYSIS ON STUDENT FEEDBACK USING MACHINE LEARNING ALGORITHMS"** being submitted by **RASNOLA GAYATHRI (207R1A05N6), VANAPARTHI ARCHANA (207R1A05P4), KUKKAMALA DEEPESH (217R5A0524)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2023-24.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr.G.Madhukar
(Associat Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on_____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **DR. G. Madhukar** Associate professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **G. Vinesh Shanker, Dr. J. Narasimharao, Ms. Shilpa, & Dr. K. Maheswari** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

RASNOLA GAYATHRI (207R1A05N6)
VANAPARTHI ARCHANA (207R1A05P4)
KUKKAMALA DEPEESH (217R5A0524)

ABSTRACT

This project uses student feedback which plays a crucial role in evaluating teaching effectiveness, curriculum relevance, and overall student satisfaction. However, the sheer volume and complexity of feedback data pose challenges in extracting meaningful insights manually. Leveraging machine learning (ML) algorithm.

Digital data trails from disparate sources covering different aspects of student life are stored daily in most modern university campuses. However, it remains challenging to (i) combine these data to obtain a holistic view of a student, (ii) use these data to accurately predict academic performance, and (iii) use such predictions to promote positive student engagement with the university. To initially alleviate this problem, in this paper, a model named Augmented Education (AugmentED) is proposed. In our study, (1) first, an experiment is conducted based on a real-world campus dataset of college students (N = 156) that aggregates multisource behavioral data covering not only online and offline learning but also behaviors inside and outside of the classroom. Specifically, to gain in-depth insight into the features leading to excellent or poor performance, metrics measuring the linear and nonlinear behavioral changes (e.g., regularity and stability) of campus lifestyles are estimated; furthermore, features representing dynamic changes in temporal lifestyle patterns are extracted by the means of long short-term memory (LSTM). (2) Second, machine learning-based classification algorithms are developed to predict academic performance. (3) Finally, visualized feedback enabling students (especially at-risk students) to potentially optimize their interactions with the university and achieve a study-life balance is designed. The experiments show that the AugmentED model can predict students' academic performance with high accuracy.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

The feedback of project scope is to encompasses a multi-faceted approach to analyzing student feedback data. It begins with the comprehensive collection of feedback from diverse sources including surveys and online platforms. A range of machine learning algorithms, including sentiment analysis, topic modeling, and classification models, are implemented to analyze the feedback Through comparative analysis, the project aims to identify the most effective algorithms for processing student feedback data. Optimization techniques such as feature selection and hyperparameter tuning are applied to enhance model performance.

1.2 PROJECT PURPOSE

The purpose of the project is to harness the power of machine learning techniques to gain valuable insights from student feedback data in educational settings By leveraging machine learning algorithms such as sentiment analysis, topic modeling, and classification, the project aims to systematically analyze the vast volume of feedback collected from students through surveys, online platforms, and other channels The primary objective is to extract meaningful patterns, sentiments, and themes embedded within the feedback data, enabling educators and administrators to understand areas of strength and areas needing improvement in teaching methods, curriculum design, and overall student satisfaction.

1.3 PROJECT FEATURES

Key features aimed at effectively analyzing and deriving insights from student feedback data. Firstly, it includes comprehensive data collection mechanisms to gather feedback from various sources, ensuring a diverse and representative dataset. the project employs advanced data preprocessing techniques to clean and prepare the collected data for analysis, addressing issues such as missing values and text normalization. it incorporates feature engineering methodologies to extract relevant attributes from the data, including sentiment scores and thematic topics, which serve as input for machine learning algorithms. decision-making processes and continuous improvement in educational practices and student satisfaction.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

System Analysis is the important phase in the system development process. Using Machine Learning Algorithms," sets the stage for understanding the significance of leveraging machine learning in educational contexts In contemporary education, gathering and analyzing student feedback is pivotal for improving teaching quality, curriculum relevance, and overall student satisfaction. However, the conventional methods of manually analyzing feedback data. This project seeks to address these challenges by integrating machine learning algorithms into the analysis process

2.2 PROBLEM DEFINITION

The primary goal of this project is to design and implement a Feedback web application that can accurately analyze the performance of each student in given dataset ,An in real-time. Specifically, the project aims to address the following key challenges.

2.3 EXISTING SYSTEM

In the given existing system of performance analysis on student feedback often relies on traditional methods, including manual evaluation and qualitative analysis. Educators typically collect feedback through surveys, interviews, or other feedback mechanisms. The data obtained is then manually categorized and analyzed, aiming to identify patterns and areas for improvement.

large volumes of feedback data is time-consuming. Educators and administrators may struggle to process the information efficiently, delaying the implementation of necessary improvements. With the growing size of educational institutions and the increasing use of online platforms for feedback collection, manual analysis becomes impractical and often fails to scale effectively to handle large datasets

2.3.1 LIMITATIONS OF EXISTING SYSTEM

- In the existing work, the system does not read Smart Card data.
- This system is less performance due to lack of prediction algorithms
- The manual nature of the existing system lacks the automation required to efficiently process and analyze large volumes of student feedback data.
- The existing system tends to provide static insights, often missing the dynamic nature of student sentiments and educational needs that can change over time

2.4 PROPOSED SYSTEM

In the proposed system, initially alleviate the challenges mentioned above, a model named Augmented Education (Augment ED) is proposed in this paper. In the proposed system, this model mainly consists of the following three modules:

Data Module in which multisource data on campus covering a large variety of data trails are aggregated and fused, and the characteristics/features that can represent students' behavioral change from three different perspectives are evaluated;

Prediction Module in which academic performance prediction is considered a classification problem that is solved by machine learning (ML)-based algorithms; and

Feedback Module in which visualized feedback is delivered individually based on the predictions made and feature analysis. Finally, Augment ED is examined using a real-world dataset of 156 college students

2.4.1 ADVANTAGES OF PROPOSED SYSTEM

- capturing a sufficiently rich profile of a student and integrating these data to obtain a holistic view
- exploring the factors affecting students' academic performance and using this information to develop a robust prediction model with high accuracy
- taking advantage of the prediction model to deliver personalize services that potentially enable students to drive behavioral change and optimize their study-life balance.
- Real-Time Feedback Analysis

2.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.5.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements:

- Processor : Pentium IV 2.4 GHz.
- RAM : 4 GB(min)
- Floppy Drive : 1.44 Mb.
- Monitor : 14’ Colour Monitor.
- Mouse : Optical Mouse.
- Ram : 512 MB.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements:

- . Operating system : Windows 7 Ultimate.
- Coding Language : Python.
- Front-End : Python.
- Designing : Html, CSS, JavaScript.
- Data Base : MySQL.

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

This project architecture shows the procedure followed for classification, starting from input to final prediction

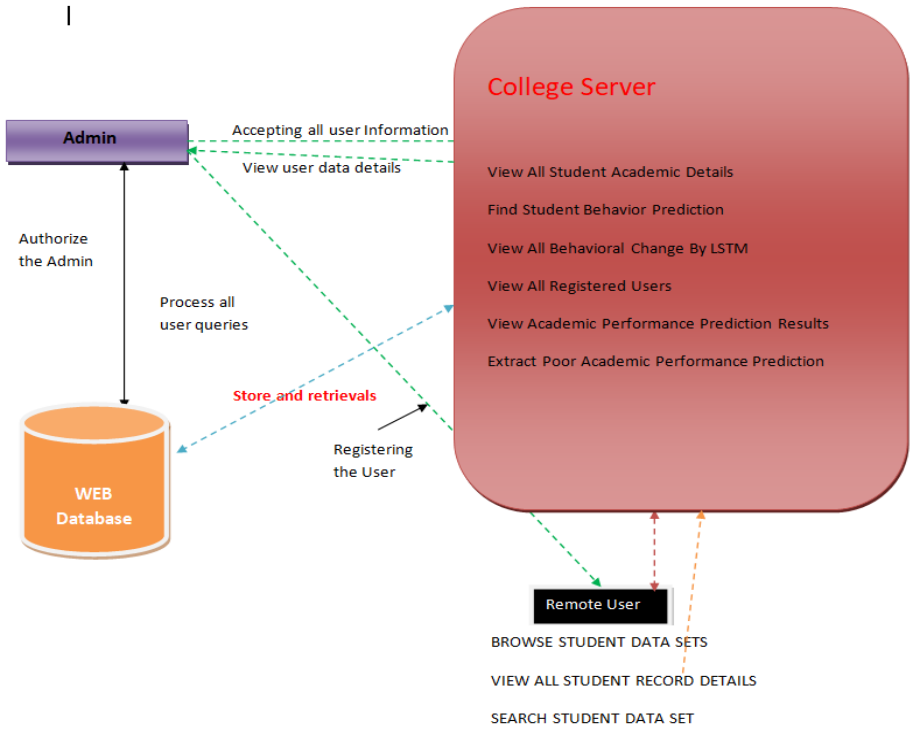


Figure 3.1: Project Architecture for Performance Analysis On Student Feedback Using Machine Learning Algorithms

3.2 DESCRIPTION

The performance analysis system for student feedback using machine learning algorithms comprises several interconnected layers designed to extract valuable insights from feedback data. Initially, the data collection layer gathers feedback from various sources, including surveys and online platforms. Subsequently, the preprocessing layer cleans and prepares the data for analysis by handling missing values and normalizing text data. The sentiment analysis layer employs natural language processing techniques to categorize feedback into positive, negative, or neutral sentiments, providing an overall sentiment overview. Topic modeling algorithms in the subsequent layer identify underlying themes within the feedback data, offering deeper insights into common issues or areas of interest raised by students. Machine learning models in the performance prediction layer analyze the feedback to predict metrics such as course ratings or teacher effectiveness.

3.3 USE CASE DIAGRAM

In the use case diagram, we have basically one actor who is the user in the trained model.

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

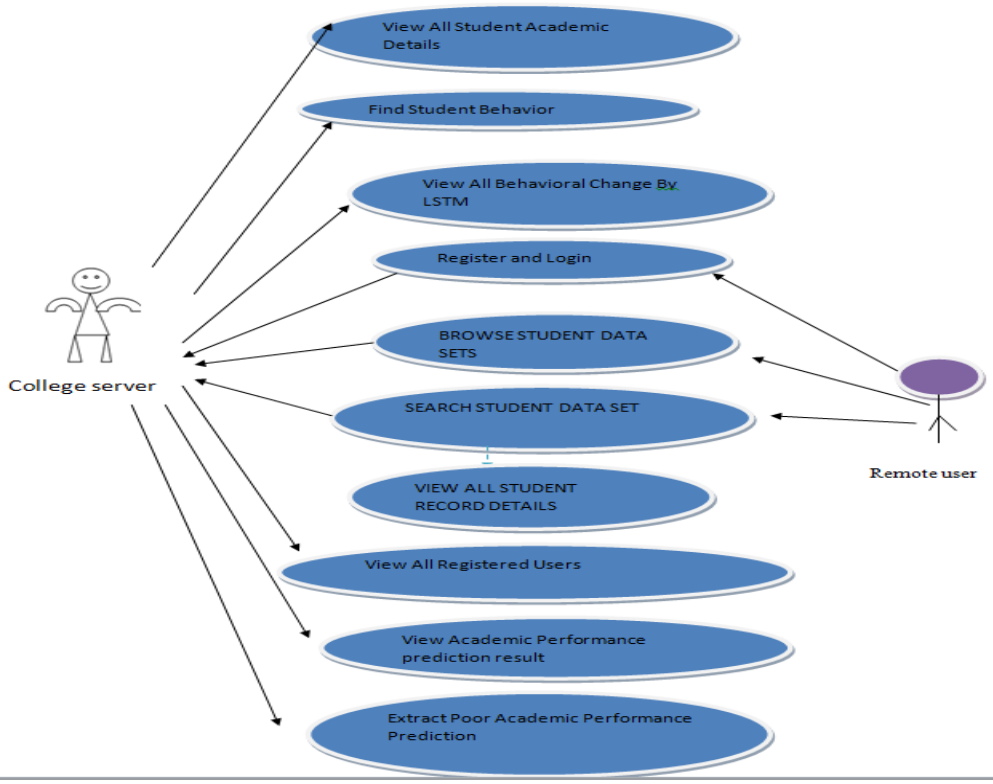


Figure 3.2: UseCase diagram for Performance Analysis On Student Feedback Using Machine Learning Algorithms

3.4 CLASS DIAGRAM

Class diagram 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 object

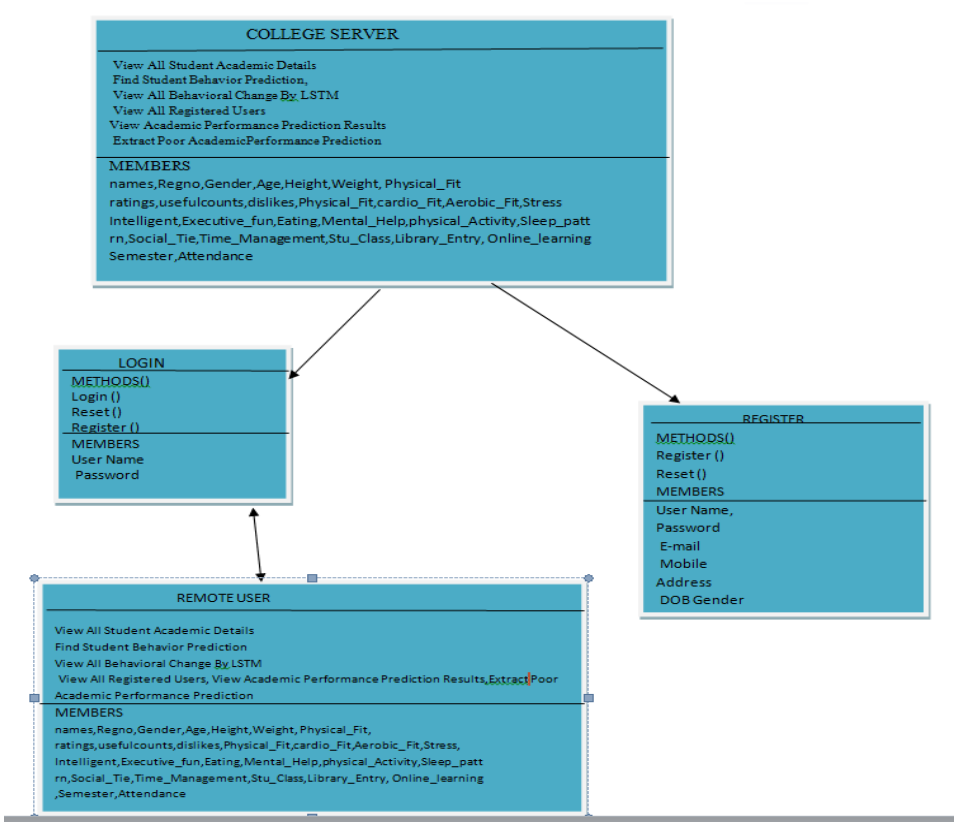


Figure 3.3: Class Diagram for Performance Analysis On Student Feedback Using Machine Learning Algorithms

3.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development.

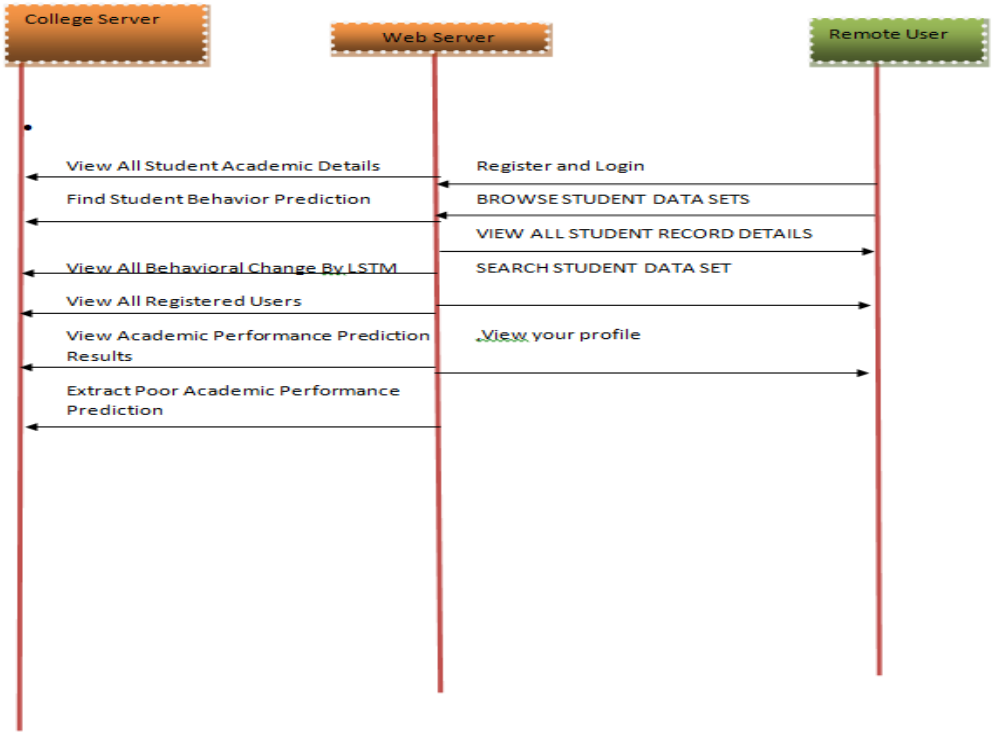


Figure 3.4: Sequence Diagram for Performance Analysis On Student Feedback Using Machine Learning Algorithms

3.6 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. They can also include elements showing the flow of data between activities through one or more data stores.

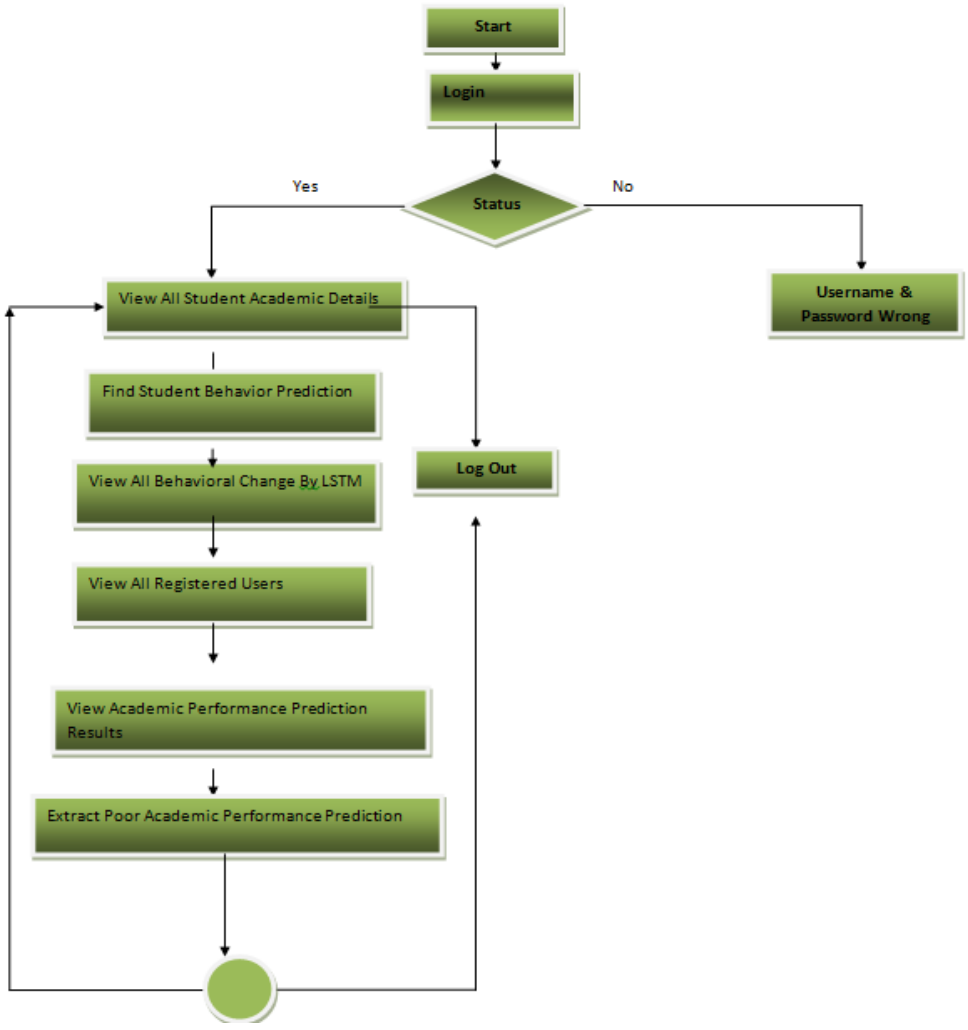


Figure 3.5: Activity Diagram for Performance Analysis On Student Feedback Using Machine Learning Algorithms

4. IMPLEMENTATION

4.1 SAMPLECODE

#Remote User

 # view.py

```
from django.db.models import Count
 from django.shortcuts import render, redirect, get_object_or_404
 import datetime
 import openpyxl
 from django.db.models import Q

      # Create your views here.
FromRemote_User.modelsimportClientRegister_Model,Clientreadings_Model,studet
data_Model


def login(request):
  if request.method == "POST" and 'submit1' in request.POST:

            username = request.POST.get('username')
            password = request.POST.get('password')
            try:

               enter = ClientRegister_Model.objects.get(username=username,
password=password)
               request.session["userid"] = enter.id
               return redirect('Add_Data_Sets')
            except:
               pass

         return render(request,'RUser/login.html')

      def Register1(request):

         if request.method == "POST":
            username = request.POST.get('username')
            email = request.POST.get('email')
            password = request.POST.get('password')
            phoneno = request.POST.get('phoneno')
            country = request.POST.get('country')
            state = request.POST.get('state')
```

```
            city = request.POST.get('city')
            ClientRegister_Model.objects.create(username=username, email=email,
password=password, phoneno=phoneno,
                                        country=country, state=state, city=city)

            return render(request, 'RUser/Register1.html')
        else:

            return render(request,'RUser/Register1.html')


    def ViewYourProfile(request):
        userid = request.session['userid']
        obj = ClientRegister_Model.objects.get(id= userid)
        return render(request,'RUser/ViewYourProfile.html',{'object':obj})

    def Add_Data_Sets(request):
        if "GET" == request.method:
            return render(request, 'RUser/Add_Data_Sets.html', {})
        else:
            excel_file = request.FILES["excel_file"]

            # you may put validations here to check extension or file size

            wb = openpyxl.load_workbook(excel_file)

            # getting all sheets
            sheets = wb.sheetnames
            print(sheets)

            # getting a particular sheet
            worksheet = wb["Sheet1"]
            print(worksheet)

            # getting active sheet
            active_sheet = wb.active
            print(active_sheet)

            # reading a cell
            print(worksheet["A1"].value)

            excel_data = list()
            # iterating over the rows and
            # getting value from each cell in row
```

```
        for row in worksheet.iter_rows():
            row_data = list()
            for cell in row:
                row_data.append(str(cell.value))
                print(cell.value)
            excel_data.append(row_data)

        studentdata_Model.objects.all().delete()
        for r in range(1, active_sheet.max_row + 1):
            studentdata_Model.objects.create(
            Regno=active_sheet.cell(r, 1).value,
            names=active_sheet.cell(r, 2).value,
            Gender=active_sheet.cell(r, 3).value,
            Age=active_sheet.cell(r, 4).value,
            Height=active_sheet.cell(r, 5).value,
            Weight=active_sheet.cell(r, 6).value,
            Physical_Fit=active_sheet.cell(r, 7).value,
            cardio_Fit=active_sheet.cell(r, 8).value,
            Aerobic_Fit=active_sheet.cell(r, 9).value,
            Stress=active_sheet.cell(r, 10).value,
            Mental_Health=active_sheet.cell(r, 11).value,
            Intelligent=active_sheet.cell(r, 12).value,
            Executive_fun=active_sheet.cell(r, 13).value,
            Eating=active_sheet.cell(r, 14).value,
            Pyhsical_Activity=active_sheet.cell(r, 15).value,
            Sleep_pattern=active_sheet.cell(r, 16).value,
            Social_Tie=active_sheet.cell(r, 17).value,
            Time_Management=active_sheet.cell(r, 18).value,
            Stu_Class=active_sheet.cell(r, 19).value,
            Attendance=active_sheet.cell(r, 20).value,
            Library_Entry=active_sheet.cell(r, 21).value,
            Online_learning=active_sheet.cell(r, 22).value,
            Semester=active_sheet.cell(r, 23).value,
            ratings=0,
            usefulcounts=0,
            dislikes=0)

    return render(request, 'RUser/Add_Data_Sets.html', {"excel_data": excel_data}

def View_Student_Records(request):
    userid = request.session['userid']
    obj = studentdata_Model.objects.all()
```

```
        return render(request,'RUser/View_Student_Records.html',{'list_objects': obj})
```

#college Server

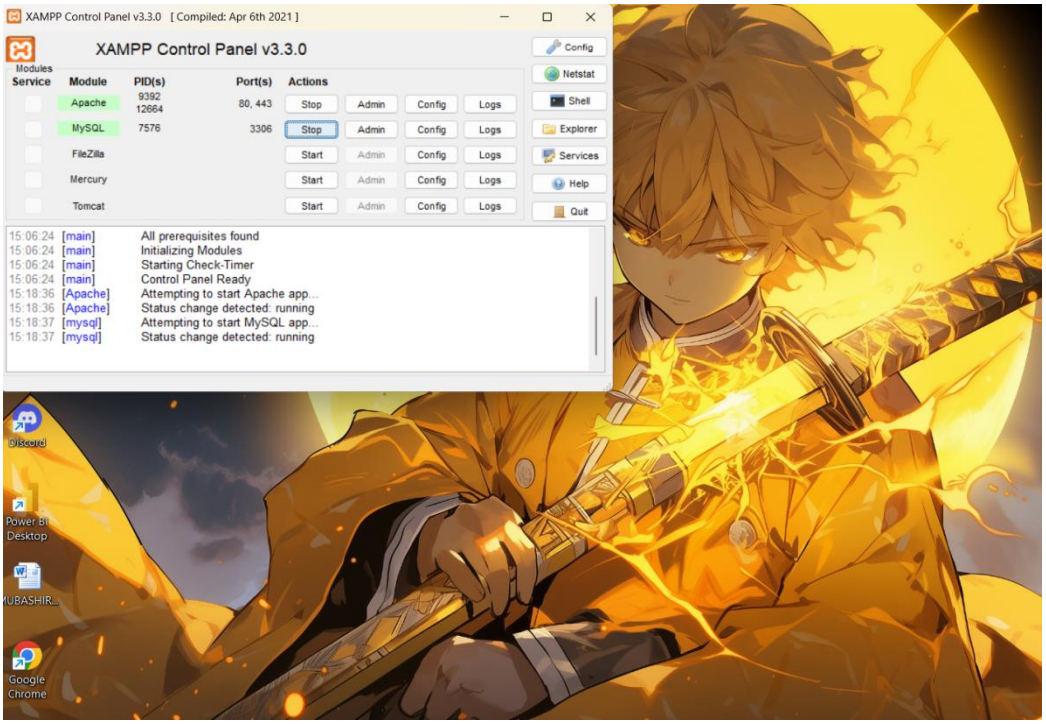
#view.py

```
from django.db.models import  Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q



# Create your views here.
from Remote_User.models import studentdata_Model,ClientRegister_Model


def collegeserverlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('admin')
        password = request.POST.get('password')
        if admin == "Server" and password =="Server":
            return redirect('viewallclients')

    return render(request,'TServer/collegeserverlogin.html')


def Find_Student_Behavior_Prediction(request):
    regno = request.POST.get('regno')
    obj1 = studentdata_Model.objects.all().filter(Regno=regno)
    return render(request, 'TServer/Find_Student_Behavior_Prediction.html',
{'objects': obj1})


def View_All_Behavioral_Change_ByLSTM(request):

    obj1=''
    btype= request.POST.get('btype')
    if btype=="Stress" :
      obj1 =
studentdata_Model.objects.values('Regno','names','Stu_Class','Gender','Age','Stress')
    elif btype=="Mental Health" :
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Mental_Health')
```

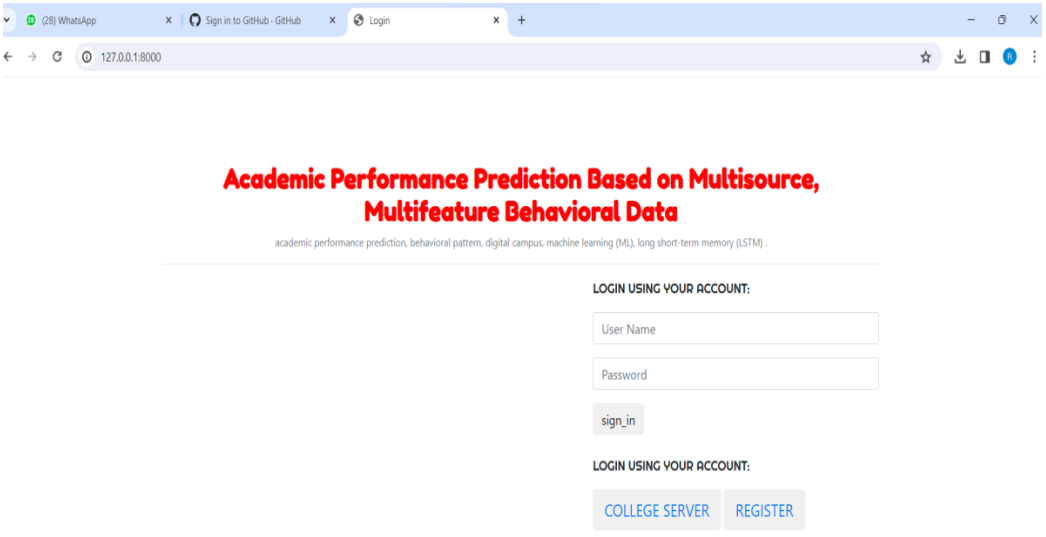
```
        elif btype == "Intelligent":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Intelligent')
        elif btype == "Executive Function":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Executive_fun')
        elif btype == "Eating":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Eating')
        elif btype == "Pyhsical Activity":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Pyhsical_Activity')
        elif btype == "Sleep Pattern":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Sleep_pattern')
        elif btype == "Social Tie":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Social_Tie')
        elif btype == "Time Management":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Time_Management')
        elif btype == "Time Management":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Time_Management')
        elif btype == "Library Entry":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Library_Entry')
        elif btype == "Online learning and Forum Discueeing":
            obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class',
'Gender', 'Age', 'Online_learning')


        return render(request, 'TServer/View_All_Behavioral_Change_ByLSTM.html',
{'objects': obj1})
    def viewallclients(request):
        obj=ClientRegister_Model.objects.all()
        return render(request,'TServer/viewallclients.html',{'objects':obj})
def negativechart(request,chart_type):
        dd = {}
        pos, neu, neg = 0, 0, 0
        poss = No
topic =
studentdata_Model.objects.values('ratings').annotate(dcount=Count('ratings')).order_b
y('-dcount')
```

```
            for t in topic:
                topics = t['ratings']
                pos_count =
studentdata_Model.objects.filter(topics=topics).values('names').annotate(topiccount=
Count('ratings'))
                poss = pos_count
                for pp in pos_count:
                    senti = pp['names']
                    if senti == 'positive':
                        pos = pp['topiccount']
                    elif senti == 'negative':
                        neg = pp['topiccount']
                    elif senti == 'nutral':
                        neu = pp['topiccount']
                dd[topics] = [pos, neg, neu]
            return
render(request,'TServer/negativechart.html',{'object':topic,'dd':dd,'chart_type':chart_ty
pe})
        def charts(request,chart_type):
            chart1 =
studentdata_Model.objects.values('names').annotate(dcount=Avg('Attendance'))
            return render(request,"TServer/charts.html", {'form':chart1,
'chart_type':chart_type})

        def dislikeschart(request,dislike_chart):
            charts =
studentdata_Model.objects.values('names').annotate(dcount=Avg('dislikes'))
            return render(request,"TServer/dislikeschart.html", {'form':charts,
'dislike_chart':dislike_chart})

        def View_College_Dataset_Details(request):
            chart = studentdata_Model.objects.all()
            return
render(request,'TServer/View_College_Dataset_Details.html',{'objects':chart})
def Extract_PoorAcademic_PerformancePrediction(request):

            objs = studentdata_Model.objects.all().filter(Q(Stress='Not Manageable'),
Q(Mental_Health='Poor') | Q(Pyhsical_Activity='No'), Q(Sleep_pattern='Late Night') )

            return
render(request,'TServer/Extract_PoorAcademic_PerformancePrediction.html',{'object
s':objs})
```

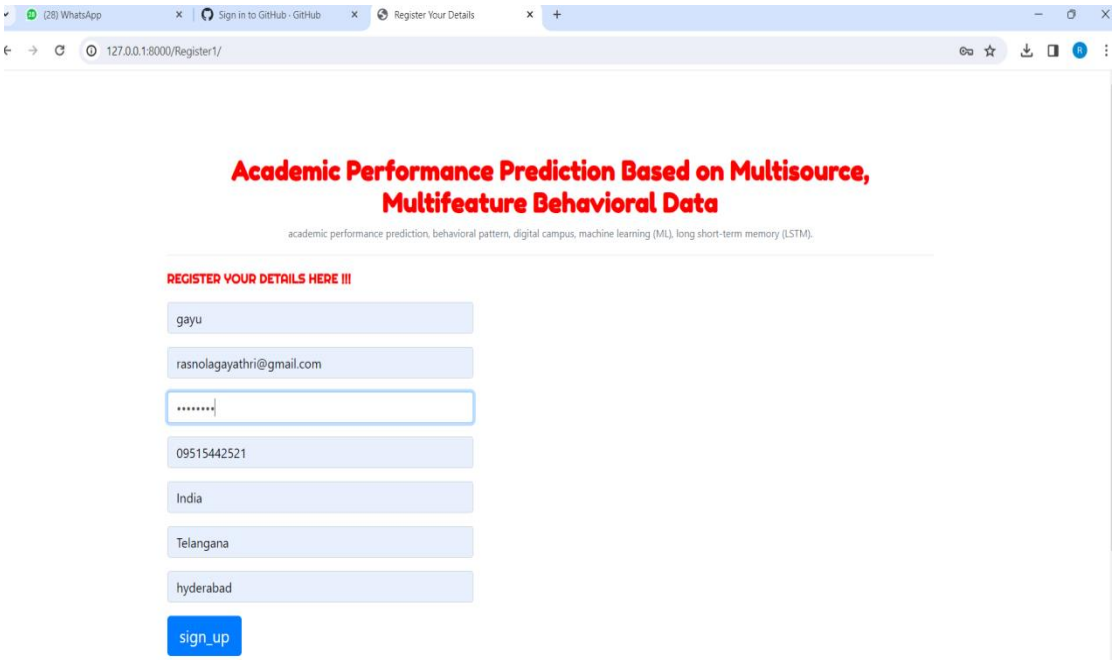
5. SCREENSHOTS



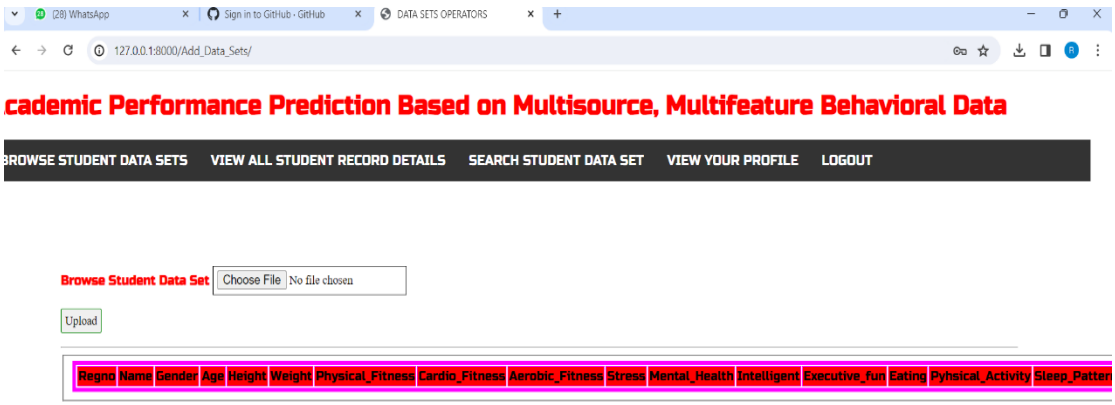
Screenshot 5.1: Open XAMP Server and start Apache and MYSQL



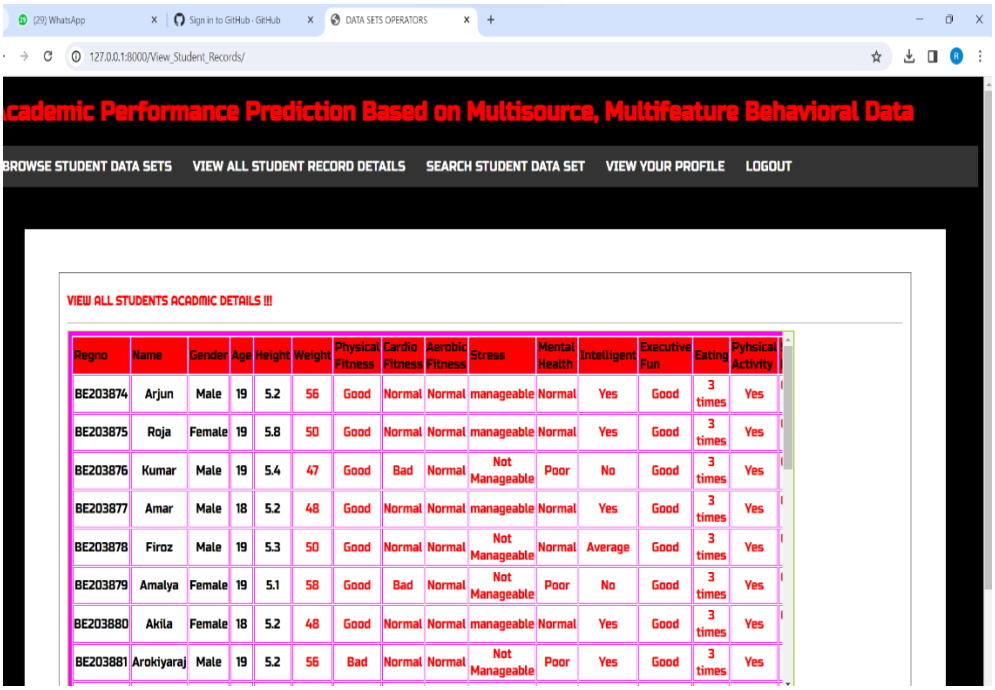
Screenshot 5.2: Login page of Remote User



Screenshot 5.3: Register as User

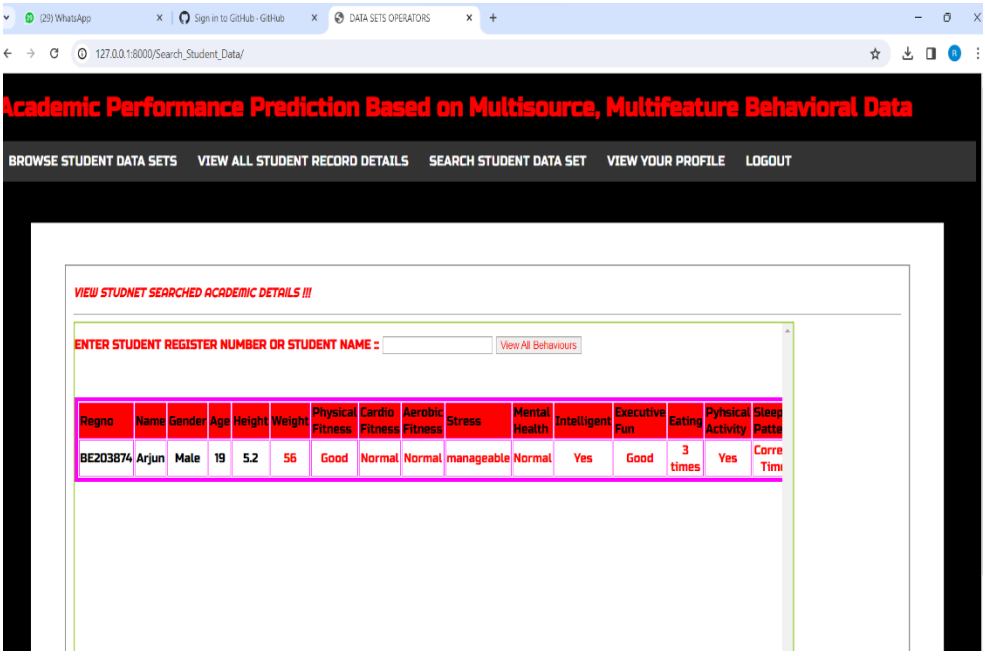


Screenshot 5.4: Select DataSet and Upload the DataSet



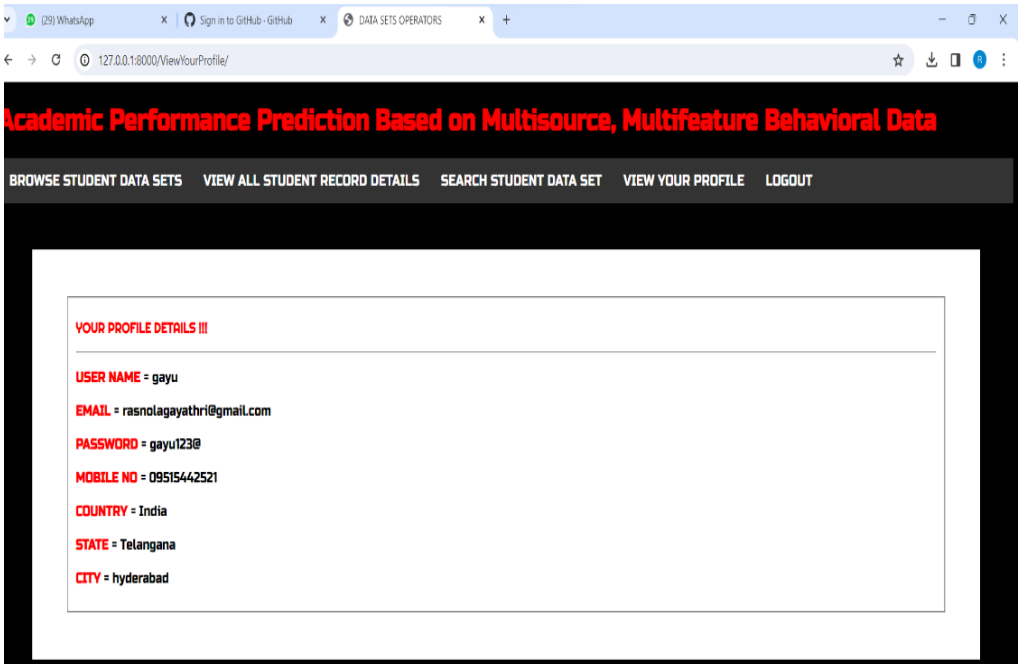
Regno	Name	Gender	Age	Height	Weight	Physical Fitness	Cardio Fitness	Aerobic Fitness	Stress	Mental Health	Intelligent	Executive Fun	Eating	Pyhsical Activity
BE203874	Arjun	Male	19	5.2	56	Good	Normal	Normal	manageable	Normal	Yes	Good	3 times	Yes
BE203875	Roja	Female	19	5.8	50	Good	Normal	Normal	manageable	Normal	Yes	Good	3 times	Yes
BE203876	Kumar	Male	19	5.4	47	Good	Bad	Normal	Not Manageable	Poor	No	Good	3 times	Yes
BE203877	Amar	Male	18	5.2	48	Good	Normal	Normal	manageable	Normal	Yes	Good	3 times	Yes
BE203878	Firoz	Male	19	5.3	50	Good	Normal	Normal	Not Manageable	Normal	Average	Good	3 times	Yes
BE203879	Amalya	Female	19	5.1	58	Good	Bad	Normal	Not Manageable	Poor	No	Good	3 times	Yes
BE203880	Akila	Female	18	5.2	48	Good	Normal	Normal	manageable	Normal	Yes	Good	3 times	Yes
BE203881	Arokiyaraj	Male	19	5.2	56	Bad	Normal	Normal	Not Manageable	Poor	Yes	Good	3 times	Yes

Screenshot 5.5: View the student record details

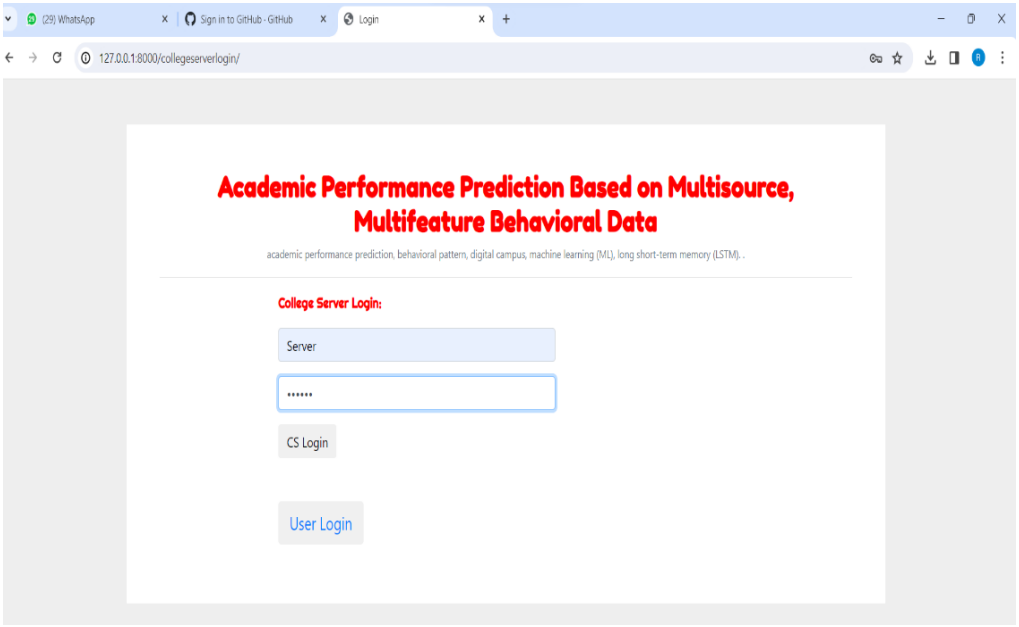


Regno	Name	Gender	Age	Height	Weight	Physical Fitness	Cardio Fitness	Aerobic Fitness	Stress	Mental Health	Intelligent	Executive Fun	Eating	Pyhsical Activity	Sleep Patte
BE203874	Arjun	Male	19	5.2	56	Good	Normal	Normal	manageable	Normal	Yes	Good	3 times	Yes	Corre Time

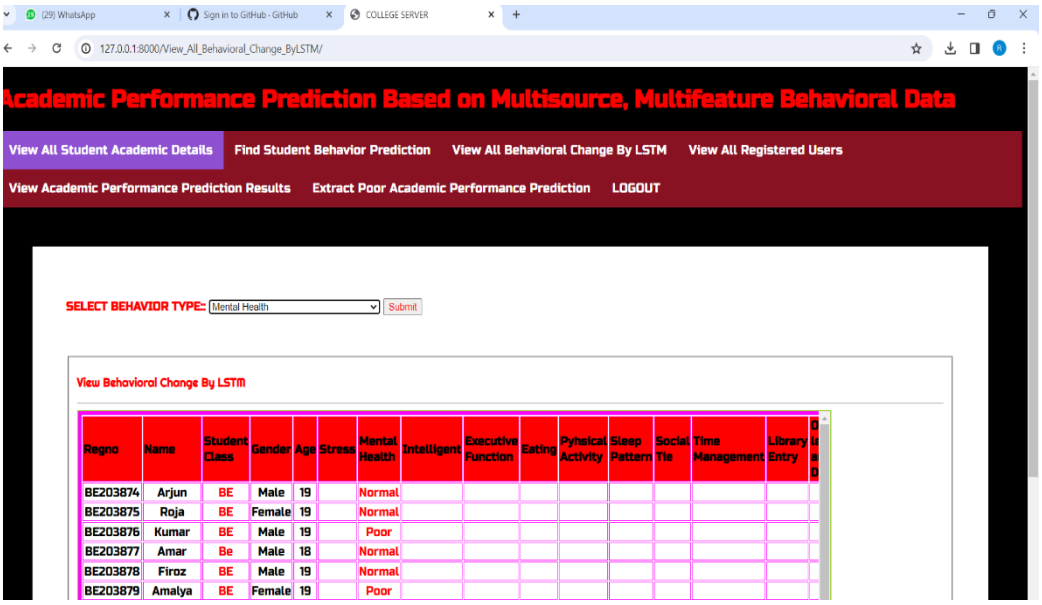
Screenshot 5.6: View the behaviour of each student through registered number



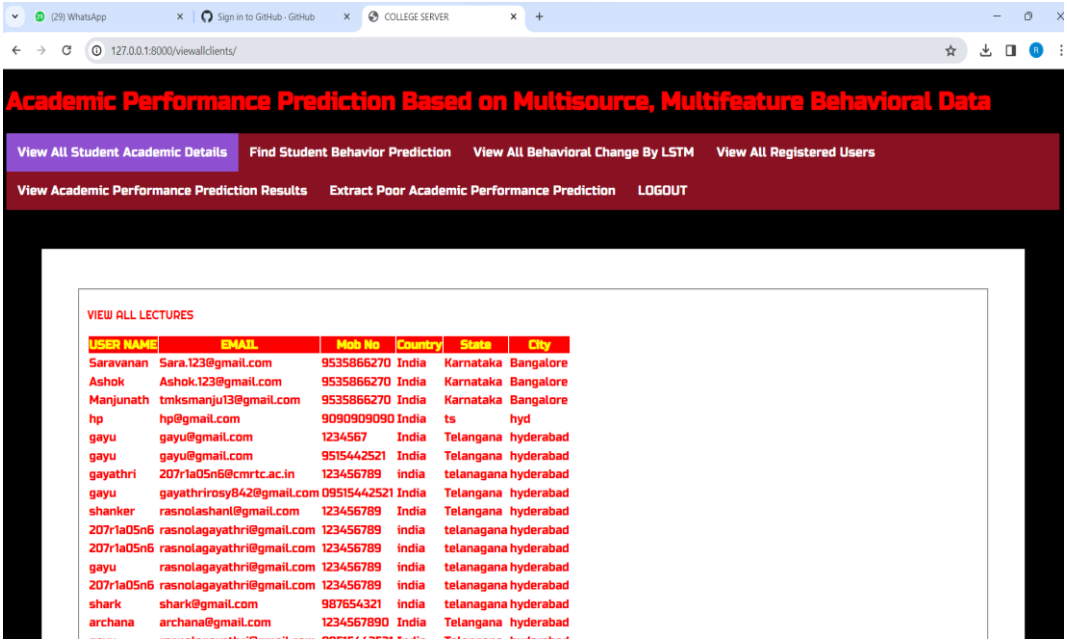
Screenshot 5.7: View the user profile



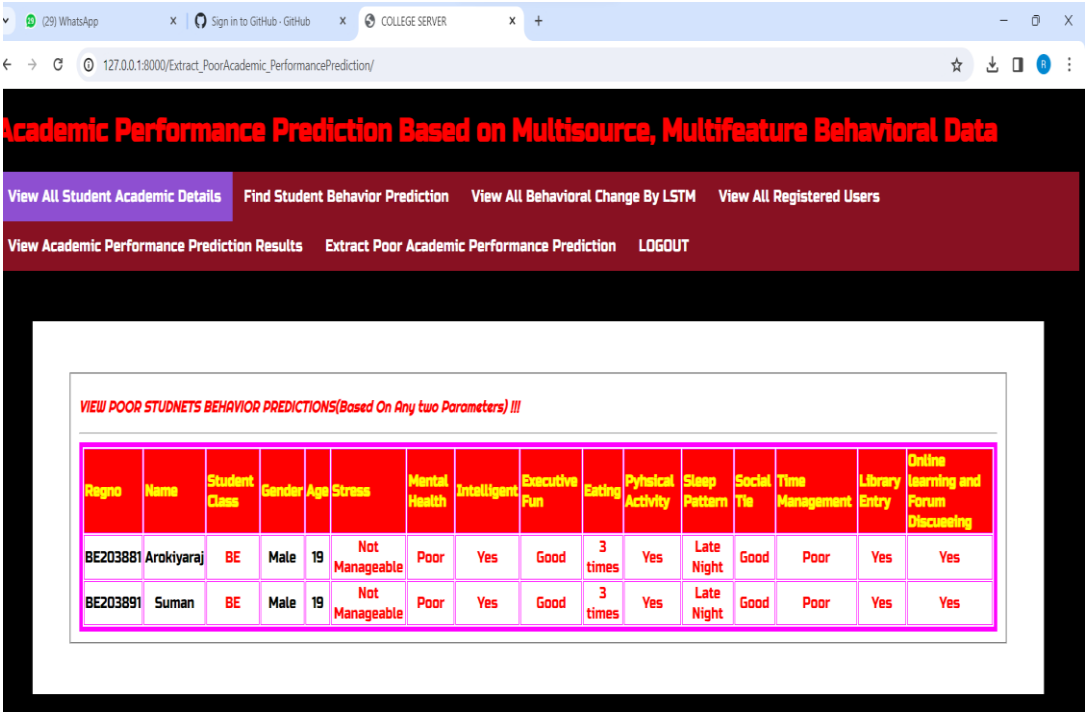
Screenshot 5.8: College server login



Screenshot 5.9: View behavioral change by LSTM algorithm

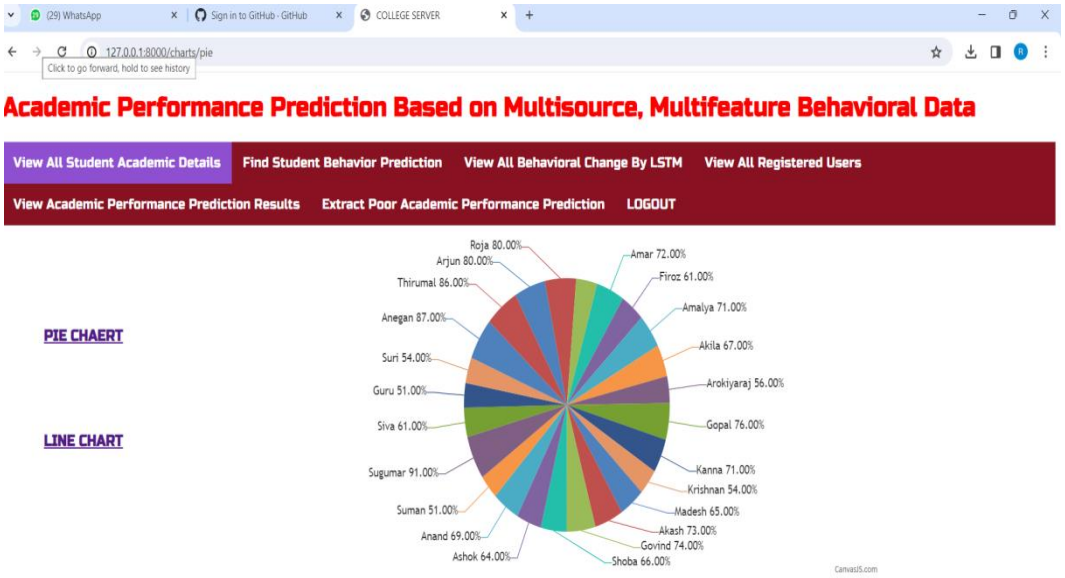


Screenshot 5.10: View all registered User

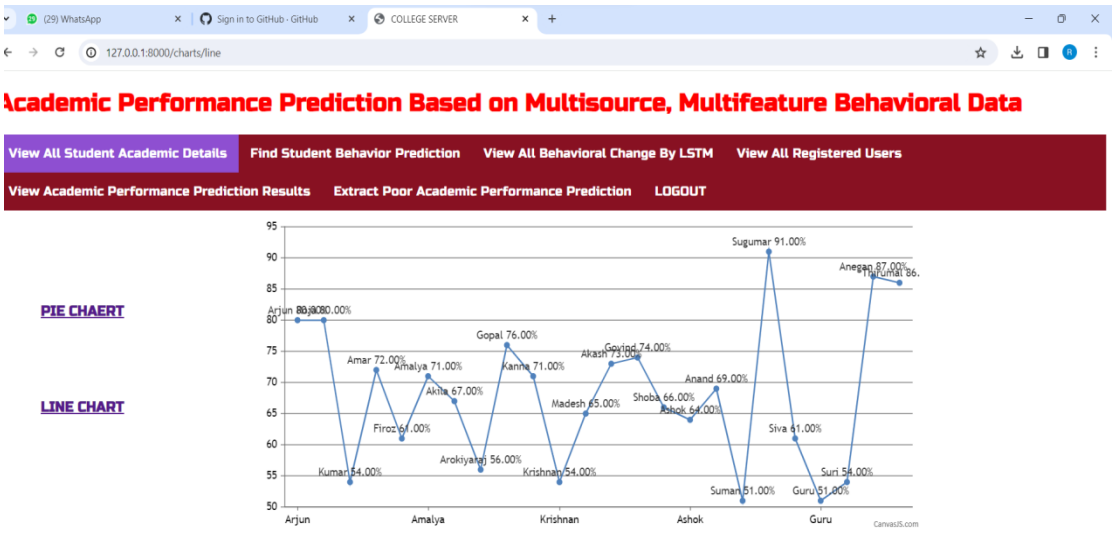


Regno	Name	Student Class	Gender	Age	Stress	Mental Health	Intelligent	Executive Fun	Eating	Pyhsical Activity	Sleep Pattern	Social Tie	Time Management	Library Entry	Online learning and Forum Discueeing
BE203881	Arokiyaraj	BE	Male	19	Not Manageable	Poor	Yes	Good	3 times	Yes	Late Night	Good	Poor	Yes	Yes
BE203891	Suman	BE	Male	19	Not Manageable	Poor	Yes	Good	3 times	Yes	Late Night	Good	Poor	Yes	Yes

Screenshot 5.11: View the Poor student behavior prediction



Screenshot 5.12: View the Academic results by Piechart



ScreenShot 5.13: View the Academic results by LineChart

6.TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration

testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Systems/Procedures	: interfacing systems or procedures must be invoked.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test Case ID	Test Case Name	Purpose	Input	Output
1	Feedback details	To view the feedback details	Feedback data	User can view the feedback details
2	Search for Feedback details By admin	To search the feedback details	Registered number or name	College server (Admin) can view performance of each student using LSTM
3	Analyse the behavioural data	To Analyse the behavioural data	behavioural data	Performance can be predicted successfully

7. CONCLUSION

7. CONCLUSION AND FUTURE SCOPE

7.1 PROJECT CONCLUSION:

we conclude the proposed system for Performance Analysis on Student Feedback using Machine Learning Algorithms represents a significant advancement in the field of educational evaluation. The innovative features and capabilities of this system address critical limitations associated with traditional manual analysis methods. The integration of machine learning algorithms introduces automation, scalability, and predictive modeling, transforming the way educational institutions interpret and act upon student feedback. It not only addresses the limitations of existing methods but also introduces a transformative approach to performance analysis on student feedback. By harnessing the power of machine learning, this system empowers educational institutions to make data-driven decisions, enhance the learning experience, and continuously improve the quality of education provided to students.

7.2 FUTURE SCOPE:

The future scope of Performance Analysis on Student Feedback using Machine Learning Algorithms holds immense potential for further advancements and applications in the educational landscape. One key area of future development lies in the refinement and augmentation of machine learning models to enhance their predictive capabilities. By incorporating advanced algorithms and techniques such as deep learning and reinforcement learning, the accuracy and granularity of performance predictions can be significantly improved. Additionally, there is a growing interest in incorporating multimodal data sources, including audio and video feedback, alongside textual feedback, to provide a more comprehensive understanding of student sentiments and experiences. personalized feedback analysis, wherein machine learning models are tailored to individual student profiles, learning styles, and preferences, allowing for more persons Machine Learning Algorithms holds promise for revolutionizing educational practices by enabling data-driven decision making processes and fostering continuous improvement in teaching and learning experiences.

8. BIBLIOGRAPHY

8.BIBLOGRAPHY

8.1 REFERENCES

[1] A. Furnham, and J. Monsen, “Personality traits and intelligence predict academic school grades," Learning and Individual Differences,
vol. 19, no. 1, pp. 0-33, 2009.

[2] M. A. Conard, “Aptitude is not enough: How personality and behavior predict academic performance,” Journal of Research in Personality,
vol. 40, no. 3, pp. 339-346, 2006.

[3] T. Chamorropremuzic, and A. Furnham, “Personality predicts academic performance: Evidence from two longitudinal university
samples,” Journal of Research in Personality, vol. 37, no. 4, pp. 319-
338, 2003.

[4] R. Langford, C. P. Bonell, H. E. Jones, T. Pouliou, S. M. Murphy, and E. Waters, “The WHO health promoting school framework for improving the health and well-being of students and their academic achievement,” Cochrane Database of Systematic Reviews, vol. 4, no. 4, pp. CD008958, 2014.

[5] A. Jones, and K. Issroff, “Learning technologies: Affective and social issues in computer-supported collaborative learning,” Computers &
Education, vol. 44, no. 4, pp. 395-408, 2005.

[6] D. N. A. G. Van, E. Hartman, J. Smith, and C. Visscher, “Modeling relationships between physical fitness, executive functioning, and
academic achievement in primary school children,” Psychology of Sport & Exercise, vol. 15, no. 4, pp. 319-325, 2014.

8.2 GITHUB LINK:

https://github.com/rasnolagayathri/PerformanceAnalysis