A comprehensive methodology for detecting stress and performance of students using machine learning

A Project Report

Submitted to the APJ Abdul Kalam Technological University

in partial fulfillment of requirements for the award of degree

in

Master of Computer Application

by

ASWIN PRABHAKARAN

TKM20MCA2013

Under the Guidance of

Dr. FOUSIA M SHAMSUDEEN

DEPARTMENT OF MCA

TKM COLLEGE OF ENGINEERING , KOLLAM

KERALA

July 2022

DEPT. OF MCA

TKM COLLEGE OF ENGINEERING KOLLAM 2020 - 22

CERTIFICATE

This is to certify that the project report entitled **A comprehensive methodology for detecting stress and performance of students using machine learning** submitted by **ASWIN PRABHAKARAN** (TKM20MCA2013), to the APJ Abdul Kalam Technological University in partial fulfillment of the M.C.A degree in Master of Computer Application is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

(Internal Supervisor) **(Head of the Department)** **(External Examiner)**

DECLARATION

I ASWIN PRABHAKARAN hereby declare that the project report **A comprehensive methodology for detecting stress and performance of students using machine learning**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Dr. FOUSIA M SHAMSUDEEN.

This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources.

I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Kollam

12-07-2022

ASWIN PRABHAKARAN

ACKNOWLEDGEMENT

I take this opportunity to express my deepest sense of gratitude and sincere thanks to everyone who helped me to complete this work successfully. I express my sincere thanks to **Dr. FOUSIA M SHAMSUDEEN**, Head of Department, MCA, TKM COLLEGE OF ENGINEERING Kollam for providing me with all the necessary facilities and support.

I would like to place on record my sincere gratitude to my project guide **Dr. FOUSIA M SHAMSUDEEN**, Professor and Head, MCA, TKM COLLEGE OF ENGINEERING for the guidance and mentorship throughout the course.

I would like to express my sincere gratitude to **Prof. VAHEETHA SALAM** , **Dr. NADERA BEEVI S** and **Prof. NATHEERA BEEVI M**, Department of MCA for their support and co-operation. I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.

ASWIN PRABHAKARAN

ABSTRACT

Nowadays, there is a big problem with mental stress, especially in young people. The time period that was formerly thought to be the most carefree is currently under a lot of stress. Today's increased stress causes a variety of issues, chief among them depression and suicide. In this experiment, we are measuring student emotional stress and assessing their academic achievement.The often-unnoticed impact of exam pressure, employment pressure, or recruitment stress on the student. The work conduct a study on how these elements impact a student's psyche and show the prevalence of stress among students.An automated system **"Student Eval"** which collects input from user and the model build using machine learning algorithm such as SVM on the dataset and model is saved.The front-end application is to gather new user features and pass them on to the stress prediction model. In addition to academic performance, clubs participation, competition participation, and achievements, teachers can also keep an eye on students accomplishments.The Teacher monitors their kids stress levels and give the right guidance and also examine the causes of stress, such as the amount of work given to the student, how long it takes them to finish it, and unfinished tasks.To identify stress and emotion in real time, a pre-trained model using CNN is used. Using opencv, a video of students is collected frame by frame. Movements of the lips and brows are used to calculate stress levels..The proposed system performs better than the existing system since most of them rely on wearable sensor data, but the proposed technique is less expensive because it uses data from a web application. A small institution or school system can use this strategy to examine the stress and academic achievement of its students.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In today's youth, depression and anxiety are common problems. Especially those who enter college right after graduating from high school. Several factors can contribute to the development of mental stress, including lifestyle changes, learning environment, academic and career pressures, different teaching styles, family work demands increase, practically have to live in a dormitory, etc. In general, the conditions mentioned above are responsible for the changes in students' lives. These effects begin to negatively impact their minds over time and cause stress. For a very long time, people believed that students would be least affected by stress or other problems. According to Masih and Gulrez (2006), stress is now recognized as a lifestyle problem that can affect anyone, regardless of developmental stage (Banerjee and Chatterjee, 2016). Students are only required to study, this should never be considered a difficult activity. The stress stemmed from parents' demands on their children, which eventually became more of a burden those children could not bear. According to National Crime Records Bureau data (Saha, 2017), one student commits suicide every hour. The office noted an 80% increase in suicide rates over a one-year period, and 1.8% of students committed suicide after failing an exam. The Lancet reported in 2012 that India has the highest suicide rate in the world for 15 to 29 year olds, and the number is not expected to decrease.

The main cause of these worrying numbers was identified as school stress. According to Lee and Larson (2000), this stress is the result of an interaction between

environmental stressors, student assessments, and their responses to these assessments. Now described as a “career stumbling block”, it has become a serious reality (Kadapatti and Vijayalaxmi, 2012). This therefore becomes a big cause for concern as it is an indication that mental health problems are on the rise in India (Nadamuri and Ch, 2011).

Some of the problems noted in students with high academic stress include depression, anxiety, behavior problems, impatience, etc. Stressed teens also have higher rates of depression, which is linked to difficulty concentrating, anxiety about future expectations, and more. (Busari, 2012). In addition, adolescents have been observed to engage in a number of risky behaviors, such as increased drug and alcohol use, unprotected sex, inactivity, eating, and drinking. irregular sleep patterns (American University Medical Association, 2009; Bennet Holloway, 2014; King, Vidourek Singh, 2014). These students were under such pressure to succeed that there were five times as many suicides. It is crucial to realize that less stressed students do not necessarily do better; conversely, they may perceive the job as boring and consider it lacking in challenges (Uchil, 2017). While certain levels of stress encourage students to do their best work, when this stress is not well managed due to lack of resources to deal with it, it can have negative effects on both students and organizations.

Regardless of the trigger, everyone experiences the same stress response. For example, stress related to marriage, exams, employment, etc. cause similar physiological responses in the body. This is mainly caused by the adrenal medullary system, which is a component of the sympathetic division of our nervous system, and the adrenal cortex axis, which triggers the ”fight or flight” response (Bourne) Yaroush, 2003). Heart rate (HR), blood pressure (BP), respiratory rate, increased blood flow to skeletal muscle, and other physiological changes in the body may be noticed. Although everyone’s response to stress can be similar, different people describe stressors differently. . The causes, sources, and effects of stressors will vary according to these differences. Multiple assignments, poor time management and social skills, peer competition, and other factors are common causes of stress in the academic environment (Fairbrother Warn, 2003). These results are in agreement with the study conducted in India and

with the results published by Sreeramareddy, Shankar, Binu, Mukopadhyay, Ray and Menezes (2007).

Other elements that are unique to each person include issues with money management, changes in the home environment, challenges juggling a busy schedule of activities outside of school, etc.

The education system also facilitates, exposing children to higher levels of stress. Too large a curriculum (Agrawal and Chahar, 2007; Sreeramareddy et al., 2007), long durations and requirements for rote learning are some of the reasons, along with crowded classes, semester grading system, and resources. inadequate and poor facilities (Awing and Agolla, 2008). (Deb et al., 2015). The fear of failure that alters children's self-esteem and confidence is frequently ingrained in them by parents and educational institutions. Ang and Huan (2006) listed increased expectations as one of the causes of higher stress levels. Since different stressors can induce different stress responses in the body, it is important to understand the multiple sources of stress in order to devise interventions that specifically target reduce students' stress levels and support their general well-being. In addition, increased stress begins to negatively impact their health and mental peace. We often see examples of students committing suicide or starting to use drugs in our daily lives. So parents, teachers, friends, college administrators, etc. take appropriate precautions. This study represents the first attempt to assess students' stress levels.
As a result, we present a system dubbed **"STUDENT EVAL"** for performance assessment and stress detection.

Chapter 2

Literature study

Although stress is clearly a big term with a significant impact, it may be controlled with small changes to our everyday routines. The majority of today's students experience stress, but it also affects housewives, managers, financiers, government employees, administrators, and politicians. To carefully address the issue and plan for successful remedies, the cause of the stress must be identified. The subjective process of stress, which is always seen as a process, encompasses a person's individual appraisal and reaction to a potentially dangerous event. Stress can lead to severe conditions like depression, anxiety, and many more.
An increase in these events is indicated by an increase in stress-related workshops, publications of various papers and research reports, as well as other indications. The rise in stress management workshops, papers, research reports, and other publications during the past few decades is indicative of the rise in incidence of stress-related illness. This study covers a variety of subjects, including the conceptual underpinnings of stress, the causes of student stress, and stress management techniques. Enhancing academic achievement is the aim. This chapter is a shortened version of my research.

A. Abou-Elfetouh.et al. [2], predicted learning outcomes of students at the beginning of a semester are one of the most important research topics in the field of educational data mining (EDM). Courses like ”Programming” and ”Data Structures” in college programs have high failure and dropout rates because students struggle with a variety of problems. To predict student performance and help them achieve better

grades in future classes, EDM is used to evaluate student data collected in different educational settings. The main objective of this paper is to study the effectiveness of deep learning in the field of EDM, especially for predicting student academic success and identifying those at risk.

J. Wang.et al. [3],proposed that Multilayer Probabilistic Probability-Based Stress Detector (MCP-SVM) is proposed to classify voices into the following categories - non-stressed, primary stress and secondary stress. The stress classifier uses a feature set that includes perceptual, MFCC, delta-MFCC and delta-delta-MFCC features. This work uses English Across Taiwan (EAT) to represent the corpus of English speech with a Taiwanese accent to observe that speakers from the same accent region tend to have similar pronunciation errors, including both word stress. The overall classification accuracy in the test data is about 84%.

F. Akhtar.et al. [4], defined Stress is an epidemic in today's world. It causes a lot of harm and is a major factor in human suicide. The main objective of this study was to evaluate the results of a stress study conducted using modern methods such as machine learning. This study created a big picture of how machine learning works in stress management. This study was conducted in several parts, including data collection using the Web of Science (WoS) database of closest keywords, designing network visualizations based on past data, evaluating evaluation of a selected research paper and conclusion of all results.

Chapter 3

Problem Statement and solution

3.1 Current System

Stress is a phrase that is usually used as a euphemism for unpleasant life events or experiences. Different perspectives on the subject are presented through logical study on pressure and unease. Stress is a natural element of human life, according to the hurried and focused lifestyles that are becoming more common. A man exhibits conduct resistances while he is adjusting to pressure. In various parts of the nation, a student commits suicide once per hour. According to a Lancet investigation(2012), a significant number of young people in our country between the ages of fifteen and twenty-nine have committed suicide. In order to prevent student suicides and other problems related to stress, it is necessary to adopt stress detection methods in schools and universities. To determine the amount of stress, numerous models and methods are available. These include real-time stress detection systems based on EEG data, web-based expert systems to identify college students' subjective stress, and understanding subjective stress in college students utilising wrist-based passive sensing.

3.2 Proposed System

There are numerous technologies and techniques for stress detection. Most of those systems need various sensors and other devices to measure the level of stress. Because of the cost of using these devices, it is not realistically practicable to monitor the stress levels of every kid in a school on a daily basis. Additionally, these systems are challenging to maintain. Here, we introduce our system for identifying and assessing student stress, known as ”STUDENT EVAL.” Exams, deadlines, returning to school, the need to balance work and study, trouble organising work, poor time management, waiting until the last minute to turn in assignments, etc. are just a few of the factors that can cause stress in students. These could push students toward suicidal thoughts or depression. The stress levels of each pupil are unknown to the teachers. Teachers can avoid accidents if they learn about the stress levels of each child in their class. While the user is logged into the system, we are developing a web application to determine their stress level, evaluate their performance and also detect their current emotions. We use factors such as club membership, wins in competitions, achievements and ratings when evaluating performance. The Work Allocated, Time Completed, Pending Tasks, and Sleep Hours metrics were included for stress detection. A pre-trained model which uses CNN to recognize real time stress and emotion is used.A video of students get captured frame by frame using opencv.The calculation of stress levels are determined using eyebrow and lip movements. The machine learning system gets trained to identify students current stress, performance levels, and emotions using these factors. The main advantage of the system is that it does not use any sensors or other equipment, which makes it less expensive and easier to maintain. To help children, teachers can track each student’s performance and stress levels.

3.3 Process Model

3.3.1 Possible Process Model

We have analysed various process models like

- Incremental model
- Agile model
- Waterfall model
- Iterative waterfall model
- Prototype model

3.3.2 Selected Model

The incremental approach is the closest approach to the project design criteria. This model requires progressive product design, implementation, and testing.

Three modules makes up the product and each is uniquely developed and built. Each module goes through four phases: requirements phase, design phase, implementation phase, and testing phase. The Admin module is the first thing created and deployed. Many administrator-controlled functions are included in the admin module.As second step, teacher module Was implemented. In addition, teachers can add students and course materials, among other features. Finally student module is implemented. The main functionality of the project, stress detection and performance assessment, can be found in the teacher module.

Chapter 4

Requirement Analysis

4.1 Main Purpose

One of the biggest problems now a days is student stress, and the generation that was previously thought to be the most carefree is now under a lot of pressure. The objective of this project is to develop a system that can recognise student stress and performance and alert professors when students require counselling or advise. The ”STUDENT EVAL” was developed using machine learning techniques. The technique is quite useful for identifying student stress.

4.2 Description of System

Students poor academic performance, inactivity, and lack of participation in extracurricular activities, among other issues, are brought on by the mental stress brought on by excessive workloads. Finding the stress of students from the data we receive from the interface and calculating the overall stress, evaluating the performance of the student in an event, and determining the present status of the student using emotion detection are some of the answers to this challenge.

4.2.1 Features of System

• Extract the necessary features.

• Training and testing of models.

• Develops an user-friendly interface.

• Connect the UI with the model.

4.2.2 Hardware and Software Requirements

• Processor : Intel Core i3/i5

• Storage : 512 GB Hard Disk space/250 GB SSD

• Memory : 8 GB RAM

• Operating System : Linux/Windows

• Platform : Python , PHP, MYSQL

• Libraries used : pandas, numpy, sklearn, Keras, Tensorflow, Opencv, joblib, pickle etc.

4.3 Project Requirements

Three modules make up this application: an admin module, a teacher module, and a student module. The application is under the overall control of the admin. Both students and teachers need the administrator's authorization to access the application. For administrators, teachers, and students, the programme has separate usernames and passwords. Teachers can enrol classes and add students once they have access, so they can do so. Teachers can assign homework to their pupils and monitor their achievement as well as their stress levels. Students can view the assignments their teachers have given them as well as their classes. The system's primary benefit is that it measures stress and performance.

4.4 Working with the system

4.4.1 Modules

Admin

Using a username and password, the administrator can access the application. The admin module has the ability to add departments, classes, subjects, and additional admin user for the application. Administrators give each teacher a special identifier and add them to the department. The distinctive identity is also added to the students.

Teacher

The user name and password for teachers allow them to access the application. Teacher adding classes inside of ”my class” (button). In the classroom, the teacher has the authority to enrol students and assign them tasks, such as assignments. Teachers can also see how stressed out and underperforming children are. Considering all these factors.

Student

The user name and password students use to access the application can be changed. Each student can examine the work that has been assigned to them by their teachers by logging into the application, and they can submit it as necessary.

4.4.2 Academic Stress and Performance detection

Within each class, teachers can see how stressed out and underperforming their pupils are. Four factors are used to calculate performance. This includes participation in clubs and groups, accomplishments, competitions, and academic grades. Work allotted, pending tasks, remaining completion time, and sleeping hours are the factors that go into the computation of stress.

4.4.3 Real time stress and emotion detection

Students can access the emotion option in their dashboard after signing into their accounts. After clicking, users are taken to a window where a webcam is used to capture a video of them. From the video captures it is distinguished frame by frame using opencv.The calculation of stress levels are determined using eyebrow and lip movements.This information is then delivered to the appropriate teacher.

4.5 Libraries and Applications used

4.5.1 For Building Machine Learning Model

Numpy

A Python library for working with arrays is called NumPy. Additionally, it has matrices, fourier transform, and functions for working in the field of linear algebra.

Pandas

Pandas is a Python library. Pandas is used to analyze data.

Sklearn

The most practical Python machine learning library is undoubtedly scikit-learn. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.In our application we use sklearn for SVM and also used for evaluating the model.

Flask

Flask is a small and lightweight Python web framework that provides useful features and tools that make creating online applications in Python easier. Since you can quickly construct a web application using only a single Python file, it allows developers flexibility and is a more approachable framework for beginning developers. Flask is used to deploy an SVM model.

Keras

Keras is a high-level open source neural network library, written in Python, capable of running on Theano, TensorFlow or CNTK. It is user-friendly, extensible and modular to facilitate faster experimentation with deep neural networks. . It supports convolutional and repeating networks separately as well as in combination.

Tensorflow

All developers can use the open source machine learning framework TensorFlow. Deep learning and machine learning applications are implemented using it. TensorFlow was developed by the Google team to explore and develop intriguing concepts in artificial intelligence. TensorFlow is a framework that is thought to be simple to comprehend because it was created in the Python programming language.

Opencv

A cross-platform package called OpenCV allows us to build real-time computer vision applications. The main areas of focus are image processing, video capture, and analytics, including face and object recognition tools.

4.5.2 Front End Development

Xampp

The World Wide Web can be used by XAMPP to serve web pages. The most important components of the package can be password protected with a specific tool. Among other things, XAMPP supports the creation and management of databases in MariaDB and SQLite.

4.6 Non-Functional Requirement Specification

Performance Requirements

- Accuracy : The system should keep its accuracy in operation and its user-friendly character.
- Speed : The system should be able to provide speed.
- Low cost: This method is both inexpensive to implement and user-friendly.
- Less Time consuming: In comparison to the existing system, it takes much less time.
- User Friendly: This proposed system is extremely user-friendly, and it allows for the creation of a pleasant environment.

Quality Requirements

- Scalability: All functional requirements will be met by the product.
- Maintainability: The system must be easy to maintain. It must keep backups in case of system failure and regularly report its activity.
- Reliability: The maximum acceptable downtime should be set as high as possible. In other words, the longer the average time between failures, the better. If the system crashes, the time required to restore the system backup should be minimal.
- Availability: This system is available because the essential software requirements are always available.
- High-functionality: Because they are so adaptable, this system is highly functional in any context.

Chapter 5

Design And Implementation

5.1 Architectural Description

There are client-side and server-side components to the architecture. The webpage that is installed on the client's computer is utilised as the user interface. The server that processes requests from clients and has the logic to deliver the right data back to them is the back-end. The database, which will permanently store all of the application's data, is also part of the backend.

Utilizing a machine learning technique, stress detection and performance evaluation are determined. At that point, developing a machine learning algorithm and train it using training data then use test data to evaluate the method.

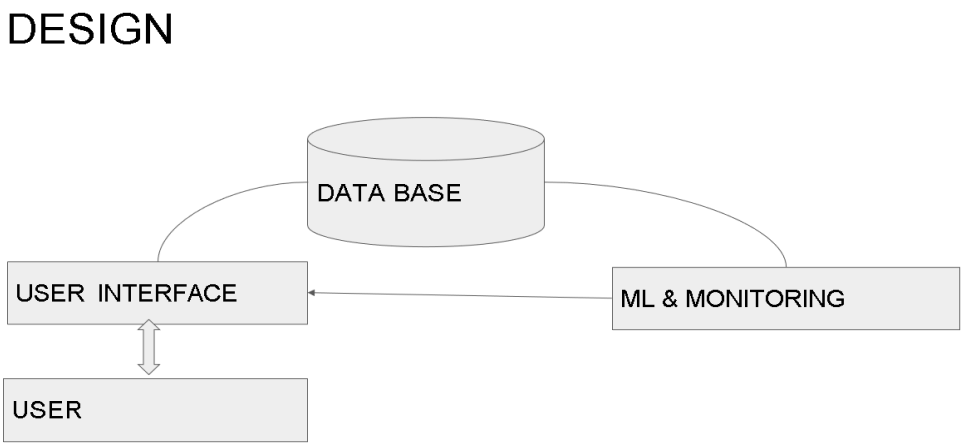


Figure 5.1: Architectural Design

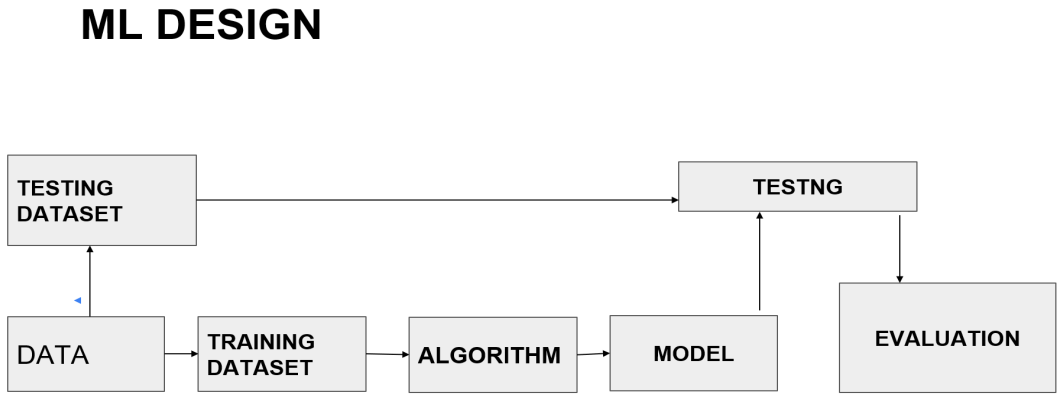


Figure 5.2: Machine Learning Design

The data from Kaggle.com is used to build the model. With two and three categories—high and low for performance prediction and low, medium, and high for stress prediction—the label column is a prediction column. The dataset has no missing values.

5.2 Methodology

This project is divided into two parts. The first step is building the model, and the second is building the user software that will interact with the built-in model. The construction of the trained model is the primary step in the automated essay grading procedure. Feature extraction, training, testing, and model evaluation are the key stages in the model generation process.

The major steps in the project are mentioned below.

- **Feature extraction:** Feature extraction is the most important part of any machine learning task and so is for it. This is a dimensionality reduction process whereby the original set of raw data is reduced into more manageable groups to process.
- **Training:**Support Vector Machine is a technique used to learn the stress detection system and evaluate student learning outcomes. It is one of the widely used supervised learning techniques for modeling binary outcomes; here low and high. Complex Neural Networks are the technique used for the detection phase of Emotions. It is one of the widely used deep learning techniques. It uses the given dataset along with the extracted features to create the model. Part of the dataset is used to train the model, while the rest is used to test the model.
- **Testing:** Tested the produced model with the remaining dataset in the testing phase. The data set is input into the produced model, and the results are saved for the model evaluation and error analysis that follows.
- **Model evaluation and error analysis:** For the error calculation, the results of

the testing data are combined with their original values. For calculating the model's efficiency, a variety of statistical measures can be used. Accuracy, precision, recall, kappa score, and other metrics are used. Proceeded with the developed model if the findings of these quantitative assessments are acceptable. If the results are bad, the model should be regenerated with more characteristics in order to achieve the optimal outcome.

The project's second phase is the user interface. The user interface was created using Python and PHP. The user is the focus of this component of the project. The input text is fed into the server in this manner. The user application also displays the returning findings. The user interface is made with simplicity and understanding in mind. Employed responsible HTML designs, including Bootstrap, CSS, and JavaScript, to provide a better user experience. User interfaces are also made using the Python Flask framework.

5.3 Decomposition Description

The System is divided into 4 modules based on the functionality of the system. That is each module is Three different users of the system.

5.3.1 Admin Module

This module contain the major contoller of the application that is the admin. Admin is the one who controls other two modules teacher and Student. Admin login to the system and can perform multiple functionalities provided for them.

5.3.2 Teacher Module

Teacher module is one of the client module.Teacher can get access to the application by registering with the application. Teachers are also provided with multiple functionalities.

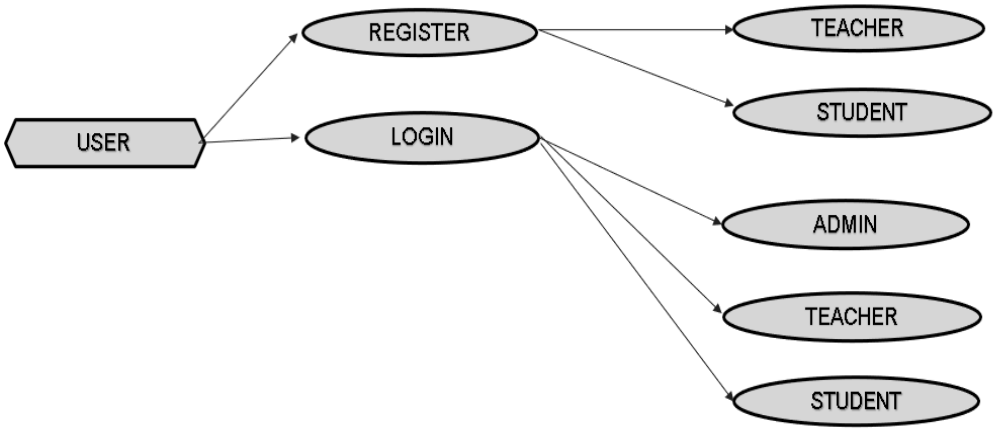


Figure 5.3: System Design

5.3.3 Student Module

Student module includes major clients of the application that is students. The main focus of application is on student stress detection, performance evaluation and emotion detection. Student can register and can use the website.

5.3.4 Server Module

All the data of each users are stored within the server module. This module provide these data to the users according to their request.

5.4 Dependency Description

The Dependencies between the modules is expressed as the data flow diagram shown below:

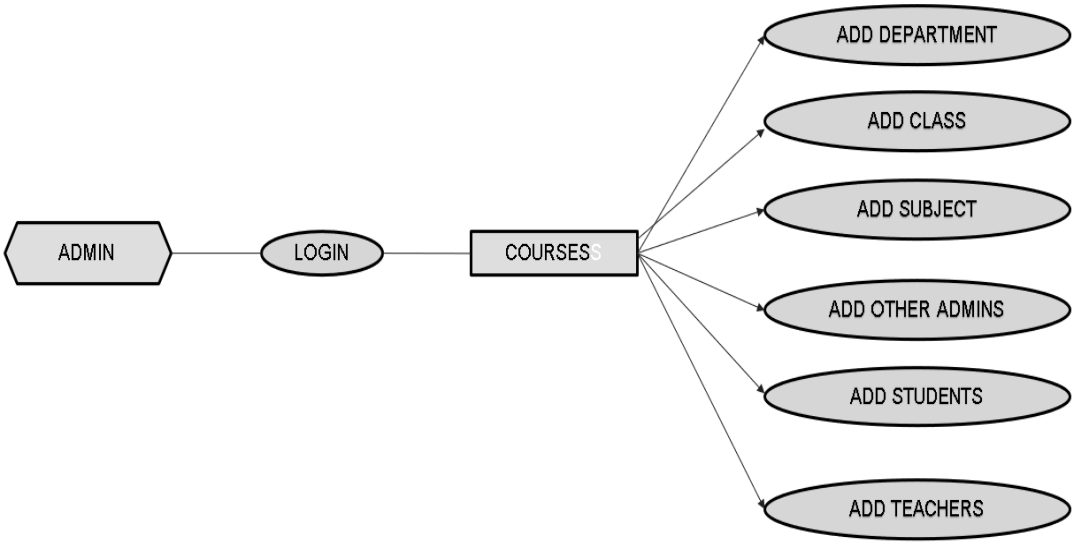


Figure 5.4: Admin

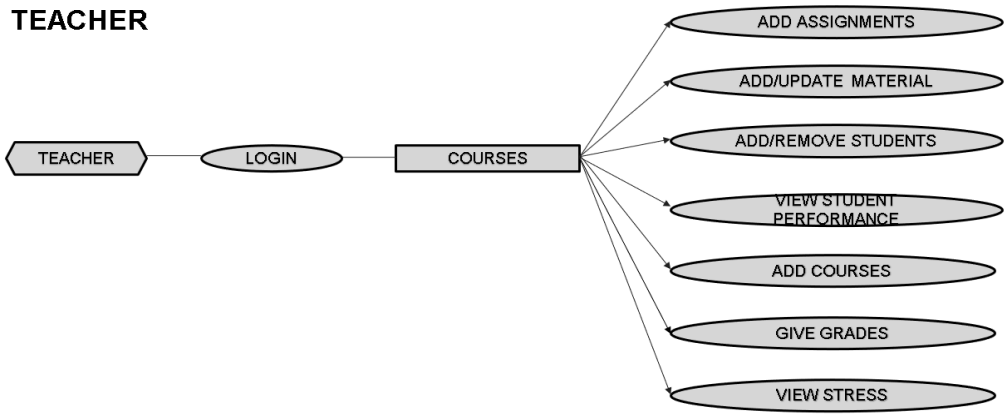


Figure 5.5: Teacher

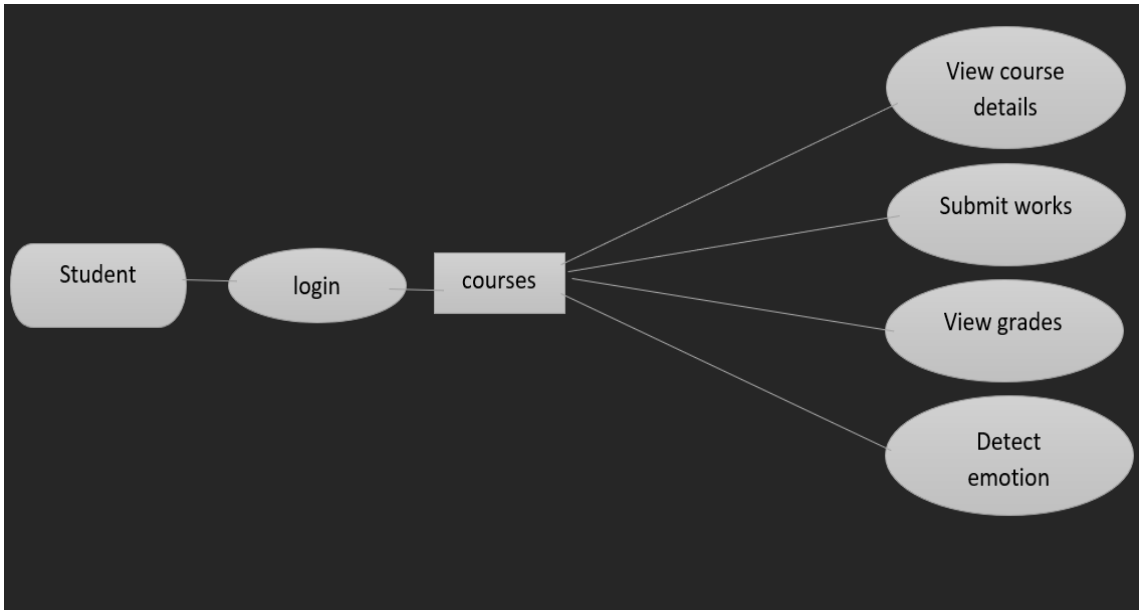


Figure 5.6: Student

5.5 Snapshots

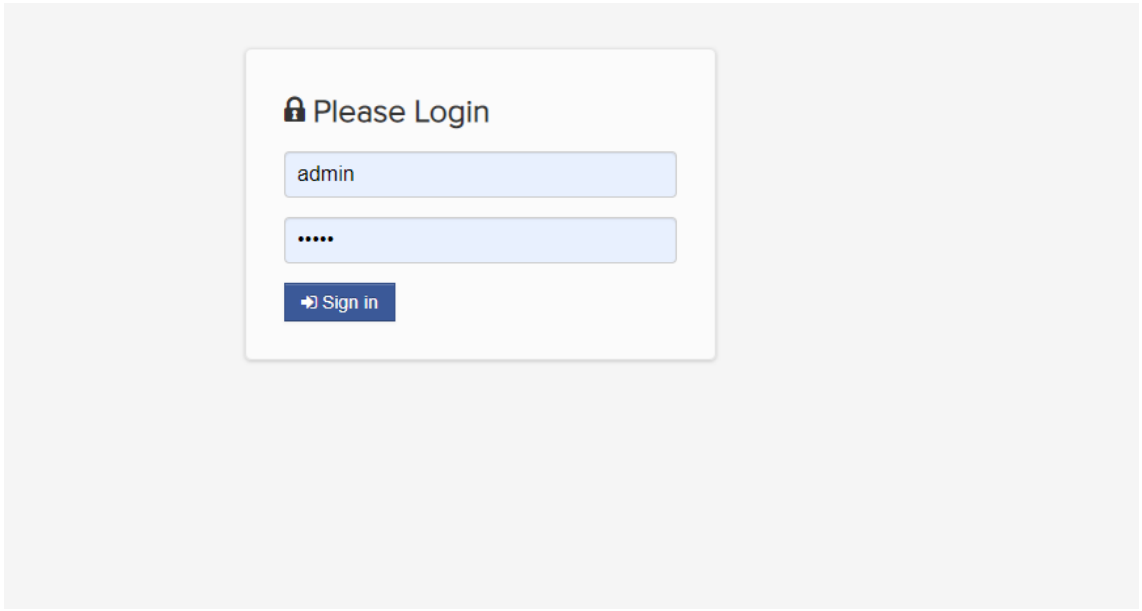


Figure 5.7: Admin login

Figure 5.8: Admin dashboard

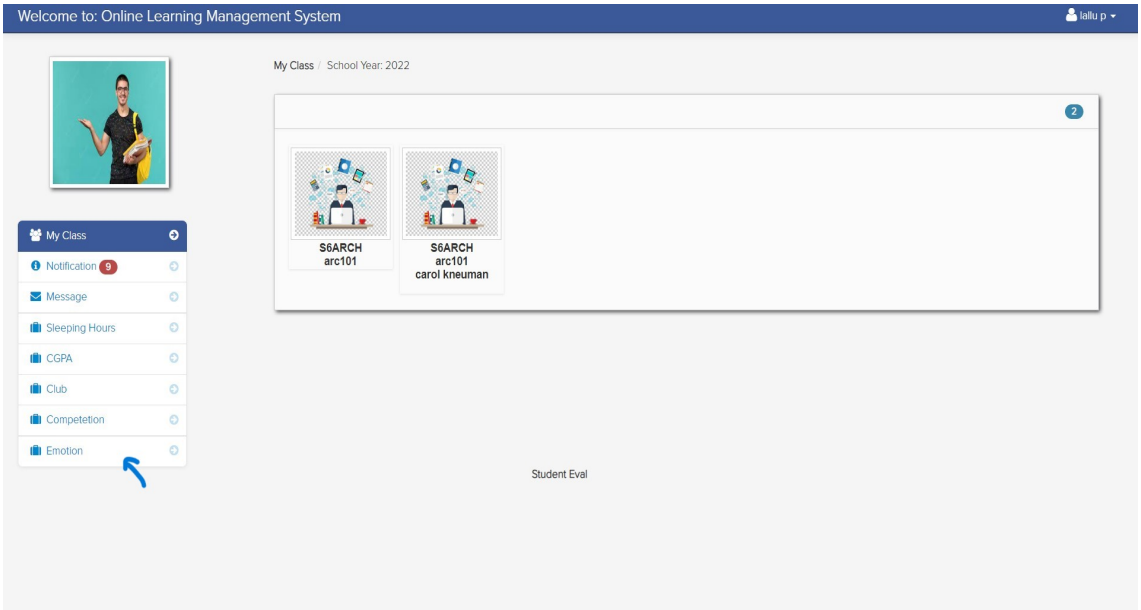


Figure 5.9: option for student emotion detecting

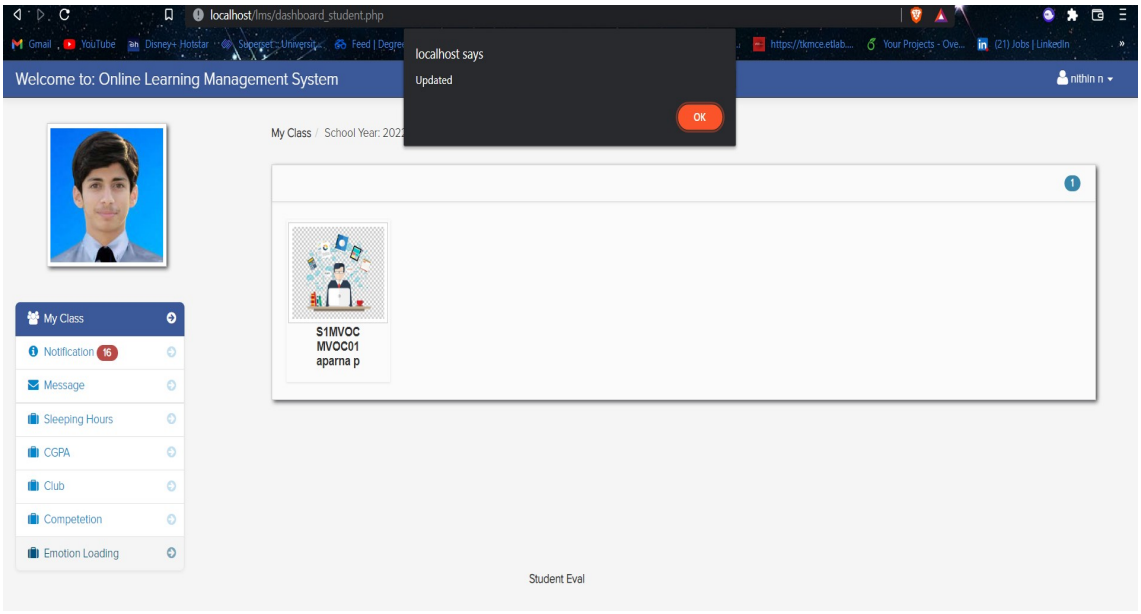


Figure 5.10: Emotion detection

Figure 5.11: Result in Tabular form

Figure 5.12: Academic Stress Result

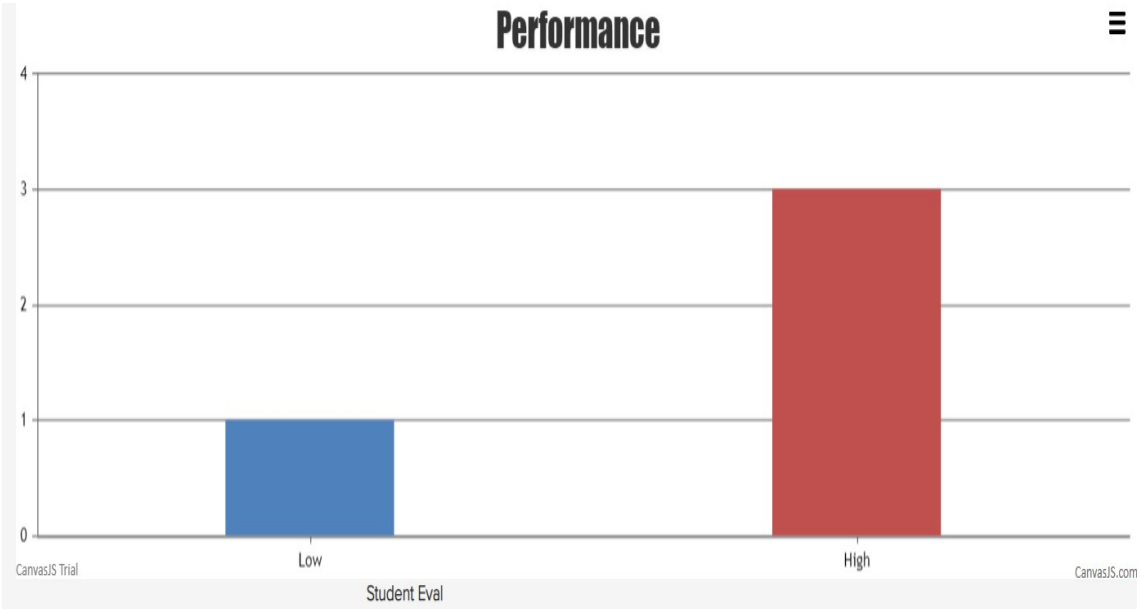


Figure 5.13: Performance Result

Chapter 6

Coding

6.1 Method for building the model

- The dataset is divided into training and testing datasets, with the training dataset being 80% of the total, and the testing dataset the remaining 20%, in order to produce the best results.
- The preprocessing stage uses the training dataset, and the features are then extracted from the preprocessed data.
- The extracted features are then examined to determine which features have the most impact on the outcomes. Next, the characteristics that have a high reliance on the Student Eval System's detection outcome.
- The Support Vector Machine is used to generate the model. Four characteristics for stress detection and performance assessment. For data that can be separated linearly, hyper-plane is used. When converting linearly unseparable data to separable data, kernel functions (linear, poly) are used.
- The generated model receives input from the testing dataset, and output is recorded.
- The effectiveness of the model is then assessed by comparing the testing dataset's real values to the results of the constructed model's evaluation of those values.

```
stress_data=pd.read_csv("stressdata.csv")
performance_data=pd.read_csv("performancedata.csv")
# stress_data1=stress_data1.dropna()
stress_data.isnull().sum()
performance_data.isnull().sum()
stress_data.shape
performance_data.shape
stress_data.head()
performance_data.head()
performance_data["Class"].value_counts()
stress_data["Intensity_level"].value_counts()
features_performance=performance_data[['Clubs or Organizations Involved', 'No of Competitions Participated',
       'Achievements', 'Academic Marks']]
x_performance=np.asarray(features_performance)
y_performance=np.asarray(performance_data['Class'])
features_stress=stress_data[['Work alloted', 'Time taken to complete the work',
       'pending work', 'sleeping hours']]
x_stress=np.asarray(features_stress)
y_stress=np.asarray(stress_data['Intensity_level'])
x_performance[0:10]
y_performance[0:10]
x_stress[0:10]
y_stress[0:10]
from sklearn.model_selection import train_test_split
x_train_performance,x_test_performance,y_train_performance,y_test_performance=train_test_split(x_performance,y_performance,test_size=0.2,random_state=4)
x_train_stress,x_test_stress,y_train_stress,y_test_stress=train_test_split(x_stress,y_stress,test_size=0.2,random_state=4)
x_train_performance.shape
x_train_stress.shape
from sklearn import svm
classifier_stress=svm.SVC(kernel='linear',gamma='auto',C=2)
classifier_performance=svm.SVC(kernel="poly",gamma='auto',C=2)
#-----------------------------------------------------------Training The Algorithm-----------------------------------------------------------------------------
classifier_stress.fit(x_train_stress,y_train_stress)
classifier_performance.fit(x_train_performance,y_train_performance)
#-------------------------------------------------------------Testing The Model--------------------------------------------------------------------------------
y_stress_predict=classifier_stress.predict(x_test_stress)
y_performance_predict=classifier_performance.predict(x_test_performance)
y_performance_predict
# y_test_performance
from sklearn.metrics import classification_report
#-----------------------------------------------------------Evaluation Of The Model----------------------------------------------------------------------------
print(classification_report(y_test_stress,y_stress_predict))
print(classification_report(y_test_performance,y_performance_predict))
```

Figure 6.1: Flask code importing pickle and json file

The model can be evaluated using a variety of statistical metrics, including Accuracy, Precision, Recall, and Kappa score etc.

6.1.1 Performance and Stress detection

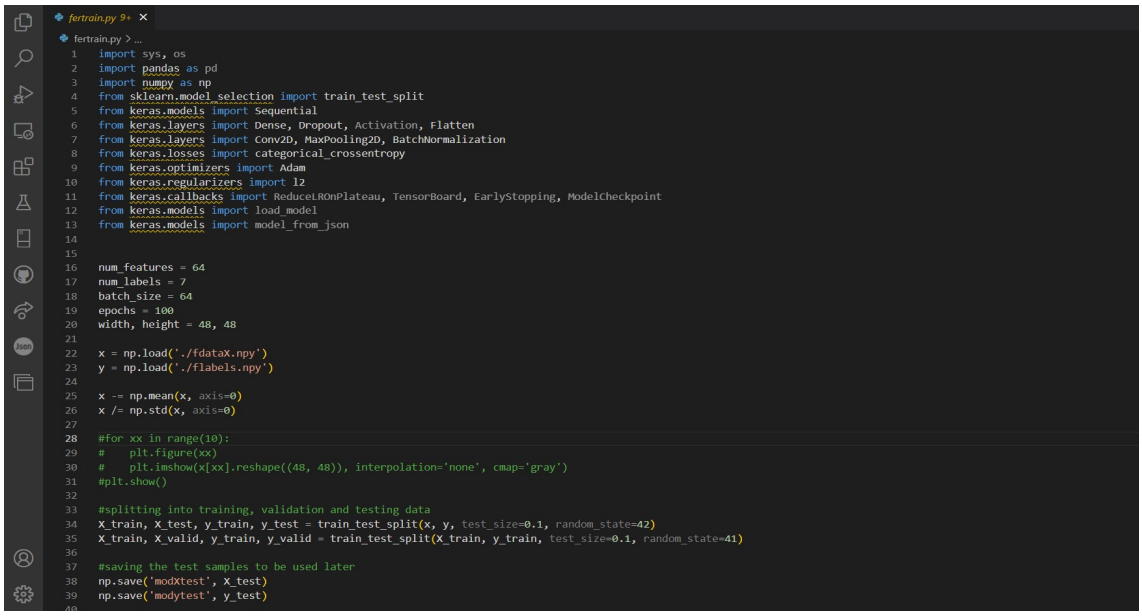
As performance evaluation criterias are clubs and organisations involved, participation in competitions, accomplishments, and academic grades.

The criterias used to indicate stress were the work assignment, completion time, amount of unfinished work, and amount of sleep.

Here, the dataset was divided into training and test data.The model is created and trained using the Support Vector Machine algorithm using training data.

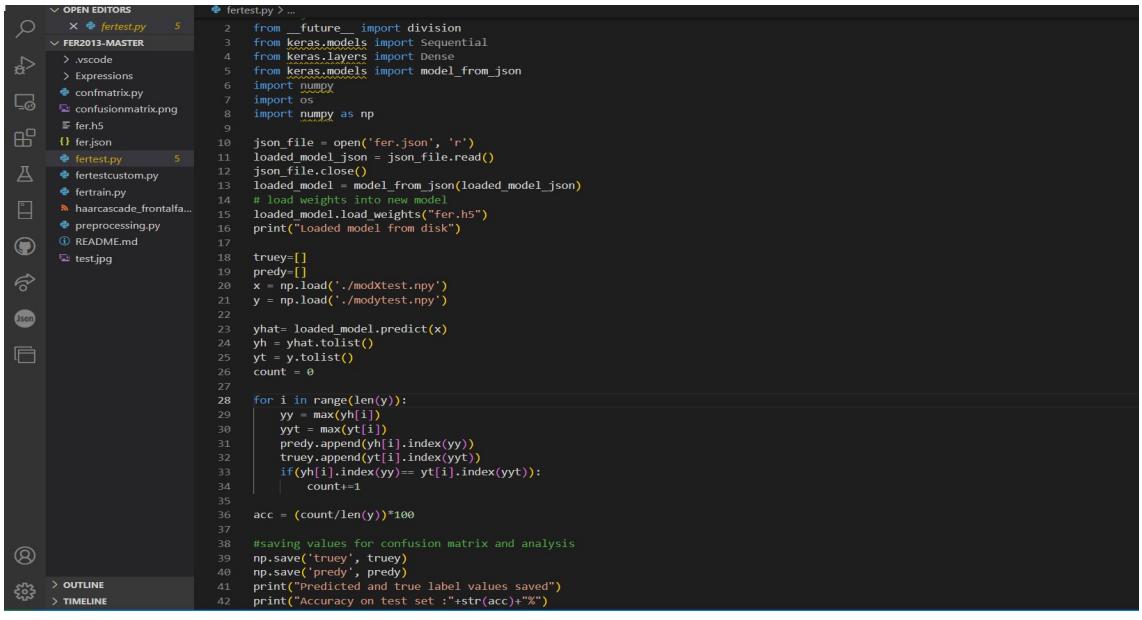
6.1.2 Emotion detection codes

The model is pretrained using the FER2013 dataset, which is utilised for emotion recognition. Model gets tested using a convolutional neural network utilising sample data (faces) from the application.



```python
import sys, os
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization
from keras.losses import categorical_crossentropy
from keras.optimizers import Adam
from keras.regularizers import l2
from keras.callbacks import ReduceLROnPlateau, TensorBoard, EarlyStopping, ModelCheckpoint
from keras.models import load_model
from keras.models import model_from_json


num_features = 64
num_labels = 7
batch_size = 64
epochs = 100
width, height = 48, 48

x = np.load('./fdataX.npy')
y = np.load('./flabels.npy')

x -= np.mean(x, axis=0)
x /= np.std(x, axis=0)

#for xx in range(10):
#    plt.figure(xx)
#    plt.imshow(x[xx].reshape((48, 48)), interpolation='none', cmap='gray')
#plt.show()

#splitting into training, validation and testing data
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=42)
X_train, X_valid, y_train, y_valid = train_test_split(X_train, y_train, test_size=0.1, random_state=41)

#saving the test samples to be used later
np.save('modXtest', X_test)
np.save('modytest', y_test)
```

Figure 6.2: training the FER2013 dataset



```python
from __future__ import division
from keras.models import Sequential
from keras.layers import Dense
from keras.models import model_from_json
import numpy
import os
import numpy as np

json_file = open('fer.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("fer.h5")
print("Loaded model from disk")

truey=[]
predy=[]
x = np.load('./modXtest.npy')
y = np.load('./modytest.npy')

yhat= loaded_model.predict(x)
yh = yhat.tolist()
yt = y.tolist()
count = 0

for i in range(len(y)):
    yy = max(yh[i])
    yyt = max(yt[i])
    predy.append(yh[i].index(yy))
    truey.append(yt[i].index(yyt))
    if(yh[i].index(yy)== yt[i].index(yyt)):
        count+=1

acc = (count/len(y))*100

#saving values for confusion matrix and analysis
np.save('truey', truey)
np.save('predy', predy)
print("Predicted and true label values saved")
print("Accuracy on test set :"+str(acc)+"%")
```

Figure 6.3: testing the FER2013 dataset

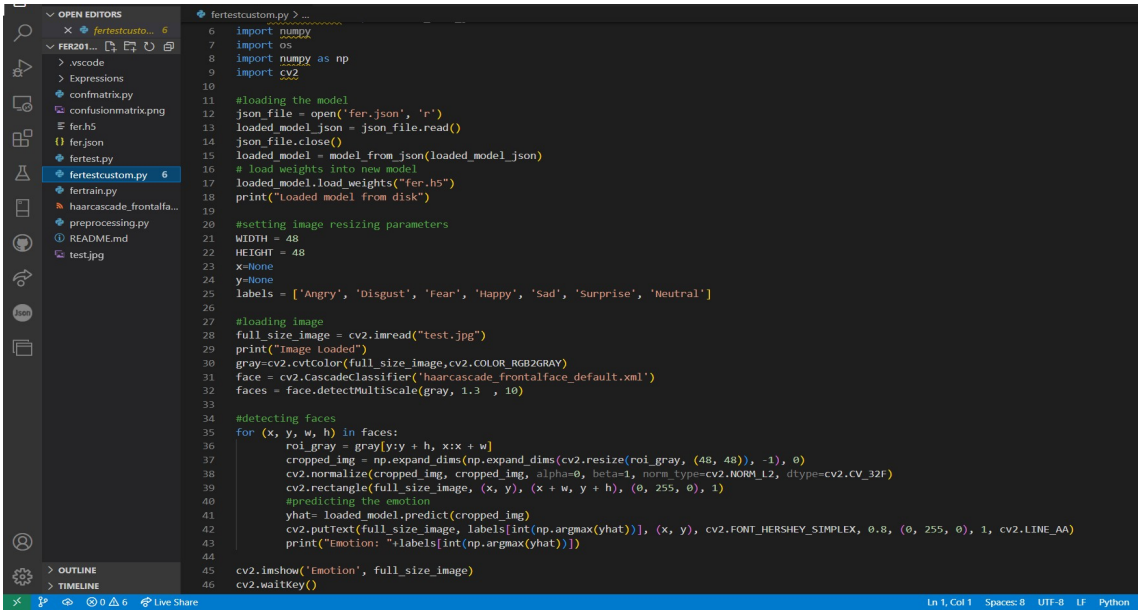


Figure 6.4: Emotion detecting

6.1.3 Model Deployment and processing codes

I used the Flask framework to install the machine learning model on a local host. From there, I loaded the model from a flask file and stored it in a pickle file to do this. After the inputs are accepted, the model will move on to the following stage of processing. The outcome of processing the front end's input is saved in a json file that is provided back to the front end. The front end's input is received via a json file.

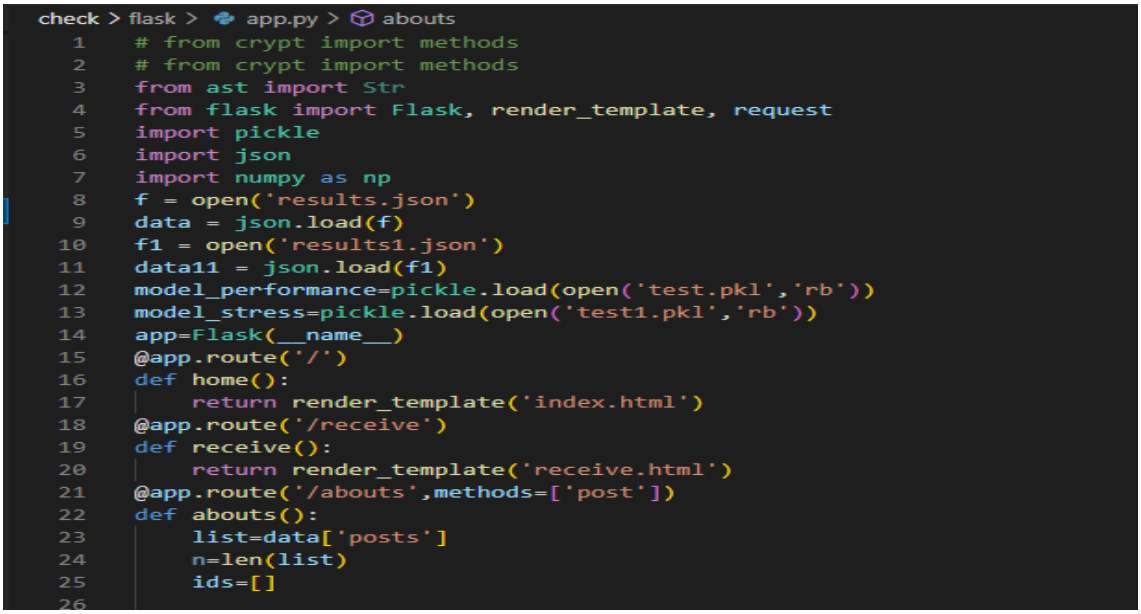


Figure 6.5: Flask code importing pickle and json file

```
check > flask > app.py > abouts
22    def abouts():
23        list=data['posts']
24        n=len(list)
25        ids=[]
26
27        for list1 in list:
28            ids.append(list1[0])
29            data1=int(list1[1])
30            data2=int(list1[2])
31            data3=int(list1[3])
32            data4=int(list1[4])
33
34
35        # data5=request.form['coi']
36        # data6=request.form['ncp']
37        # data7=request.form['ach']
38        # data8=request.form['am']
39            arr1=np.array([[data1,data2,data3,data4]])
40        # arr2=np.array([[data5,data6,data7,data8]])
41            pred_stress=model_stress.predict(arr1)
42            ids.append(pred_stress[0])
43            #ids[list1[0]]=pred_stress[0]
44        # pred_performance=model_performance.predict(arr2)
45
46        list=data11['posts']
47        n=len(list)
```

Figure 6.6: model operation code in flask

```
33  <ul  id="da-thumbs" class="da-thumbs">
34          <?php
35                  $my_student = mysqli_query($conn,"SELECT * FROM teacher_class_student
36                  LEFT JOIN student ON student.student_id = teacher_class_student.student_id
37                  INNER JOIN class ON class.class_id = student.class_id where teacher_class_id = '$get_id' order
38                  while($row = mysqli_fetch_array($my_student)){
39                  $id = $row['teacher_class_student_id'];
40                  $sid=$row['student_id'];
41                  $sleep=$row['sleep']/1;
42                  $class1=$row['class_id'];
43                  $query1=mysqli_query($conn,"SELECT COUNT(*) FROM assignment where class_id='$get_id' ")or die(m
44                  $row1=mysqli_fetch_array($query1);
45                  $work=$row1[0]/1;
46                  $query2=mysqli_query($conn,"SELECT SUM(time_taken) FROM student_assignment where student_id='$s
47                  $query3=mysqli_query($conn,"SELECT COUNT(*) FROM student_assignment where student_id='$sid' ")c
48                  $row2=mysqli_fetch_array($query2);
49                  $row3=mysqli_fetch_array($query3);
50                  $complete=$row3[0];
51                  $timetake=$row2[0]/1;
52                  if($complete!=0)
53                  $timetake=$timetake/$complete;
54                  $pending=$work-$complete;
55                  $club=$row['club'];
56                  $cgpa=$row['cgpa'];
57                  $na=$row['na'];
58                  $ac=$row['ac'];
59                  $posts1[] = array(  $sid,$club,$na, $ac,$cgpa);
60                  $response1['posts'] = $posts1;
61                  $posts[] = array(  $sid,$work,$timetake, $pending, $sleep);
62                  $response['posts'] = $posts;
63
64                  $fp = fopen('C:\xampp\htdocs\lms\check\flask\results.json', 'w');
```


Figure 6.7: passing input data as json file from the front end

Chapter 7

Testing and Implementation

7.1 Testing and it's types used

The main task following software development is to determine whether the experimental results and the actual results agree. Testing is the process in question. It is employed to ensure that the created system is free from errors. The basic objective of testing is to find errors and missing operations by running the software. Additionally, it makes sure that the developer satisfies all of the project's goals. The goal of testing is to identify defects in the developed software as well as ways to increase its correctness, usability, and efficiency. It seeks to gauge a software program's performance, functionality, and specification. The developed programme is put through tests, and the outcomes are compared to the required documentation. Debugging is carried out when there are too many faults that have happened. After debugging, the software is once more tested to make sure there are no errors. Unit testing, integration testing, and system testing are the main testing methodologies used in this project.

- In unit testing, tested to each distinct piece of software. It ensures that the software's many components all function as intended.
- The integrated separate pieces are tested in integration testing to see if the intended function was achieved or not. It assists us in identifying any flaws that might occur when the units are merged.
- During system testing, the complete piece of software is examined to make sure

it satisfies all the criteria. The testing method that took place while this project, ”Student Eval” was being developed is described in the tables below. This outlines the numerous processes that were used to produce the project without errors.

7.1.1 Unit Testing

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Data collected from Students	Process student user input data and output is obtained	Same as expected	Pass
2	Detection of student stress	Train data and detect it.Correct output obtained.	Same as expected	Pass
3	Detection of student performance	Train data and detect it.Correct output obtained.	Same as expected	Pass

Table 7.1: Unit test cases and results

7.1.2 Integration Testing

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Model outputs the correct result based on the input.	The output of the model should be based on the data detection which is collected from user.	Same as expected	Pass

Table 7.2: Integration test cases and results

7.1.3 System Testing

Model	Input	Expected result	Actual result	Pass or Fail
Performance	Clubs : 9 Competition : 8 Achievements : 7 Academic Marks : 8	high	high	Pass
Stress	Work Allotted : 7 Time Taken : 2.16 Pending Work : 1 Sleeping Hours : 5	low	low	pass

Table 7.3: System test cases

Chapter 8

Results and Discussion

Using machine learning model SVM to sight stress and performance was the foremost objective of the study. The system additionally appears to hold out every task as supposed. consistent with this model, the system determines whether or not a Student includes a high or poor performance level and, similarly, whether or not they square measure beneath low, medium, or severe stress.

8.1 Performance analysis for validation

Performance analysis is finished to spot the simplest model having the best detection rate. the overall analysis metrics like Accuracy, Precision, Recall, F1 score and confusion Matrix square measure used. High accuracy here indicates the improved detection rate and reduced warning rate.

The foremost necessary performance matrix is precision. precision is that the fraction of relevant instances among the retrieved instances of classification techniques.

Precision = T P/(T P+FP)

Recall is additionally one among the foremost common performance metrics to estimate the performance of the classification model.The recall is that the share of all relevant examples that were truly found by the classification rule to be the retrieved instance.

Recall = T P/(T P+TN)

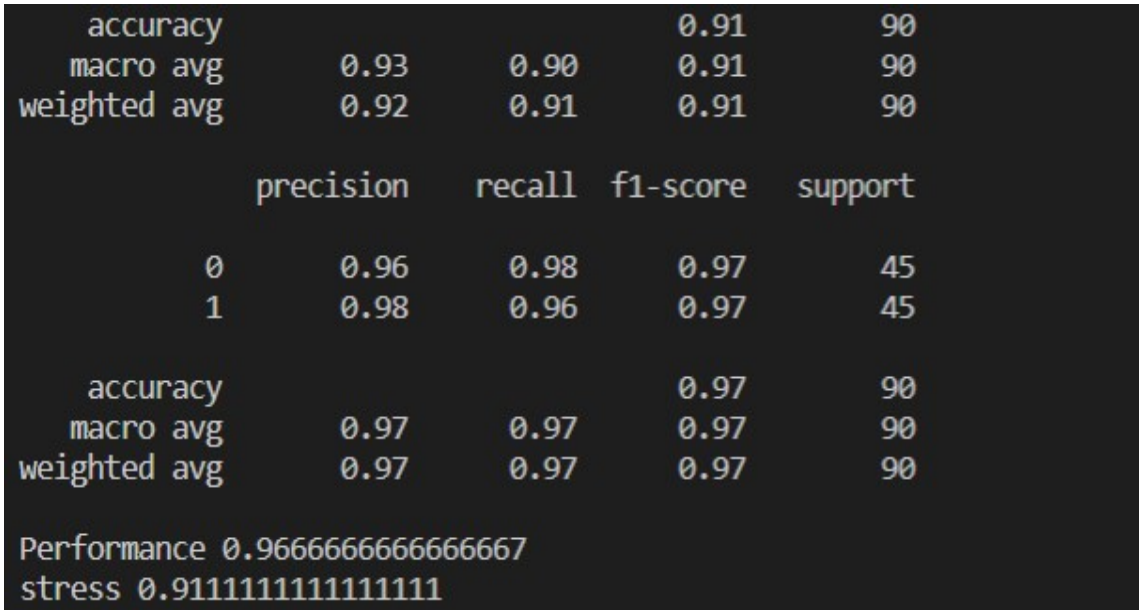


Figure 8.1: Performance matrix of SVM model

The f1 score may be a metric for evaluating accuracy in applied math analysis of binary classification. The f1 score is calculated by taking under consideration the classification algorithm's exactness and recall.

F1score = Precision Recall/(Precision+ Recall)

8.2 Confusion Matrix

A confusion matrix is a type of graph that is frequently used to illustrate how well a classification model (also known as a classifier) performs on a set of test data for which the true values are known. It makes it possible to see how well an algorithm is working. It makes it simple to spot class labelling confusion, like when one class is frequently labelled with another. The confusion matrix is used to generate the majority of performance metrics.

The following terms are contained within the confusion matrix:

- True Positive (TP): Observation is positive and should be positive.
- True Negative (TN): Observation is negative and should be negative.
- False positive (FP): Observation was negative, however foreseen to be positive.

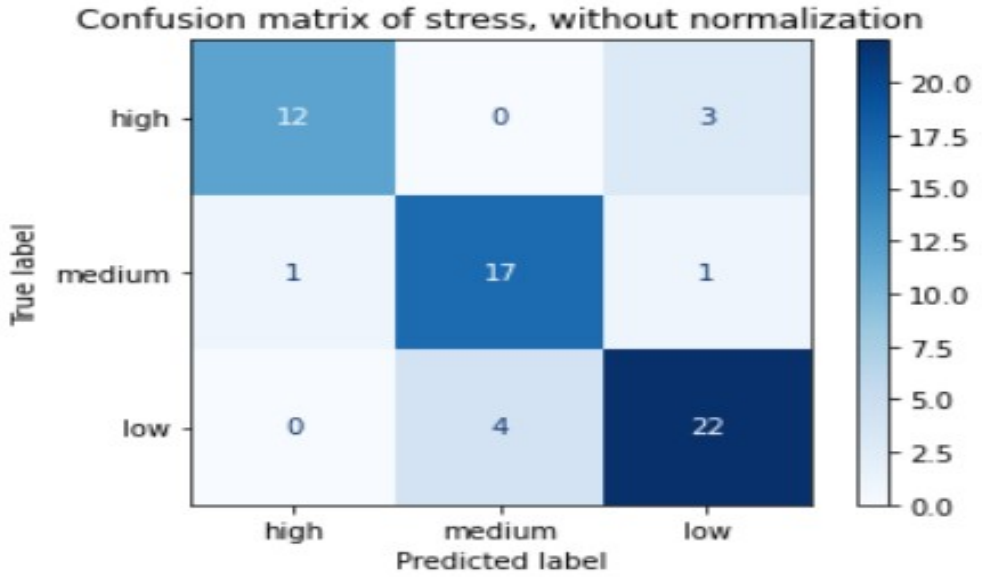


Figure 8.2: confusion matrix of stress(without normalization)

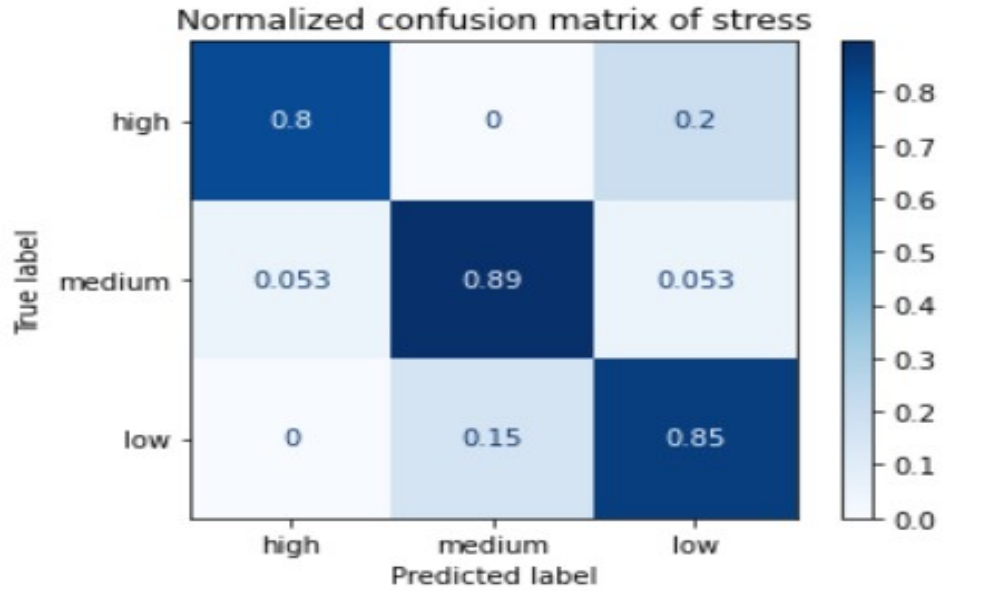


Figure 8.3: confusion matrix of stress(with normalization)

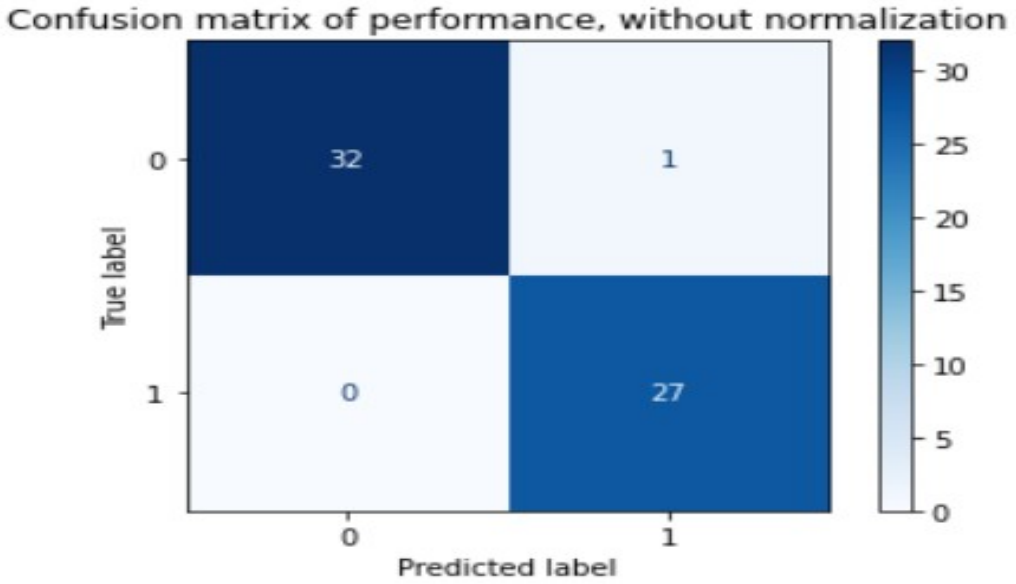


Figure 8.4: confusion matrix of performance(without normalization)

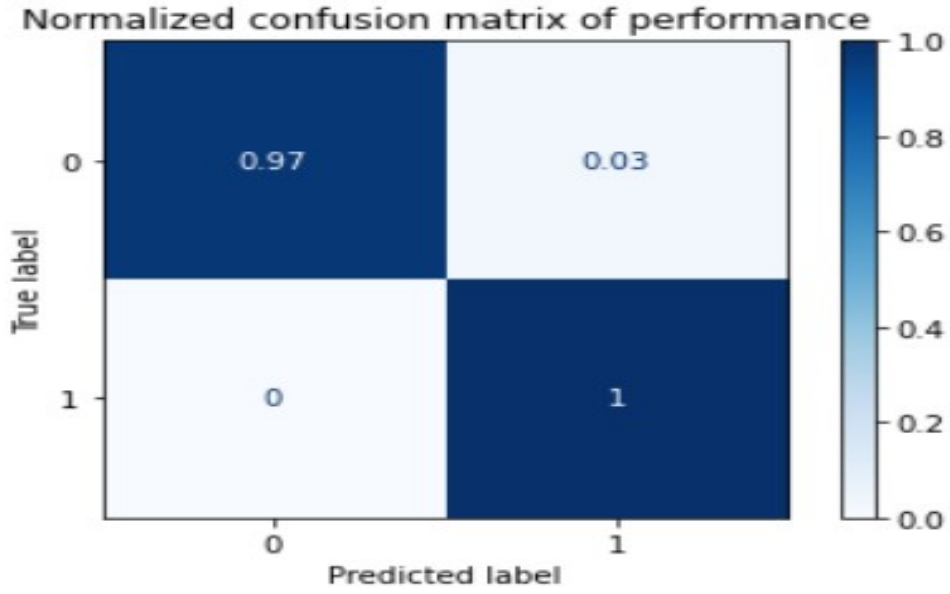


Figure 8.5: confusion matrix of performance(with normalization)

- False negative (FN): The observation is positive, however the prediction is negative.

8.3 Benefits and Drawbacks

The suggested method is a machine learning model with 96.6 percent performance accuracy and 92 percent stress detection accuracy. Over the current system, the proposed system has greater benefits. The suggested system significantly reduces time. Like every other system, this one has drawbacks of its own. However, they are minor in comparison to the benefits, and they can be fixed in the future.

8.3.1 Benefits

• Stress detection in children and adolescents is now essential. This tool makes it simpler to gauge pupils' levels of stress and their academic achievement across the board within a facility.

• Not utilising any stress detection sensors. to make this application affordable and simple to maintain.

• The user interface is created in a way that makes it simple for the user to use, indicating that the system's front end is user friendly.

8.3.2 Drawbacks

• Because the sleeping hour is only measured once a week, there is a potential that the value may be off. As a result, we need to develop an alternate way to measure the sleeping hour accurately, or this parameter will need to be updated in the future.

• It is not possible to send a direct SMS message to the parents.

Chapter 9

Conclusion and Future Work

The system, ”Student Eval,” is a very effective machine learning application that evaluates a student's general performance, emotional condition, and level of stress in the given situation. It can be accomplished in a variety of ways. A machine learning model was created in this case to identify the performance. Results from the model seem promising and show opportunity for further improvement. The error rate of the machine learning model is incredibly low. Python was used primarily in the project's construction. It uses a flask server to connect to the user interface built with HTML, JavaScript, and CSS.

To assess performance and pinpoint student stress, Student Eval is used. This tool allows teachers to monitor their students levels of stress and performance. so that the teachers can help the students perform while also helping them to manage their stress. Teachers can upload group assignments, add students to their courses, and publish their courses using this programme. My application provides universities and institutions with an easy-to-use website as a result.

This specific machine learning model can be expanded to include other dimensions in the future. Future research on this subject might make schools safer for students. Future updates may include advice for teachers on how to support or encourage the low-performing, extremely stressed pupils on a weekly basis.Some other features, such as an academic calendar, the ability to upload attendance, the recommendation to students who experience more problems to see doctors, and some extra features like an expert system for real-time stress detection can be added.

References

[1] B. Egilmez, E. Poyraz, W. Zhou, G. Memik, P. Dinda, and N. Alshurafa, "Ustress: Understanding college student subjective stress using wrist-based passive sensing," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 673–678.

[2] A. Nabil, M. Seyam and A. Abou-Elfetouh, "**Prediction of Students' Academic Performance Based on Courses' Grades Using Deep Neural Networks,**" in IEEE Access, vol. 9, pp. 140731-140746, 2021, doi: 10.1109/ACCESS.2021.3119596.

[3] J. Wang, G. Chang, J. Wang and S. Lin, "**Stress Detection Based on Multi-class Probabilistic Support Vector Machines for Accented English Speech,**" 2009 WRI World Congress on Computer Science and Information Engineering, 2009, pp. 346-350, doi: 10.1109/CSIE.2009.739.

[4] F. Akhtar, M. B. Bin Heyat, J. P. Li, P. K. Patel, Rishipal and B. Guragai, "**Role of Machine Learning in Human Stress: A Review,**" 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 2020, pp. 170-174, doi: 10.1109/ICCWAMTIP51612.2020.9317396.

[5] J. Wijsman, R. Vullers, S. Polito, C. Agell, J. Penders, and H. Hermens,**"Towards ambulatory mental stress measurement from physiologicalparameters,"** inProc. Humaine Assoc. Conf. Affect. Comput. Intell. Inter-act., Sep. 2013, pp. 564–569.

APPENDIX

Snapshots(continuation)

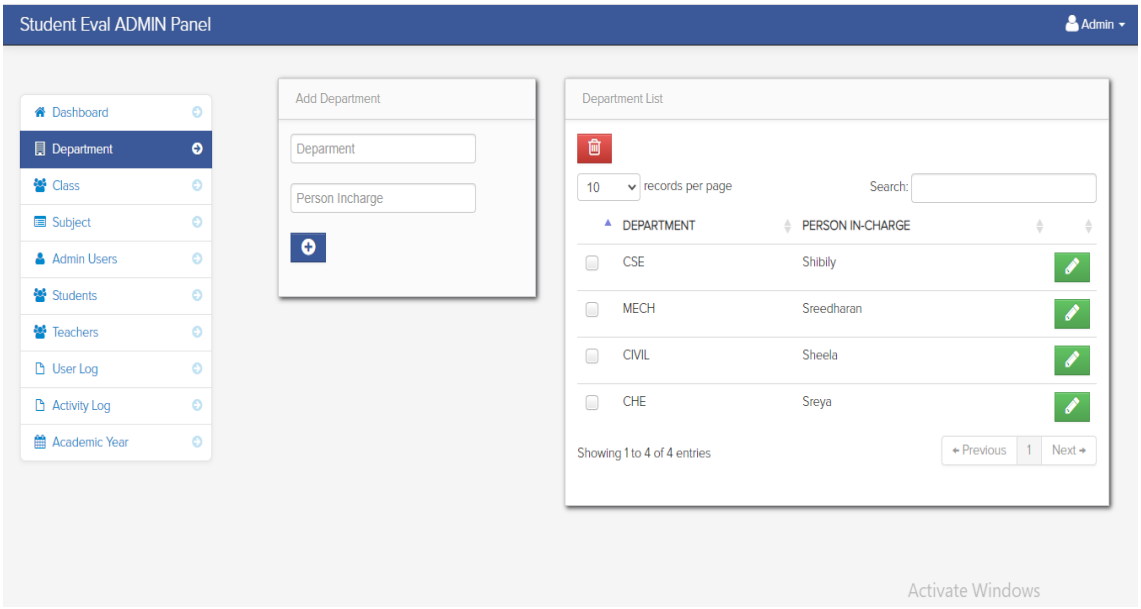


Figure 9.1: Admin add and view department

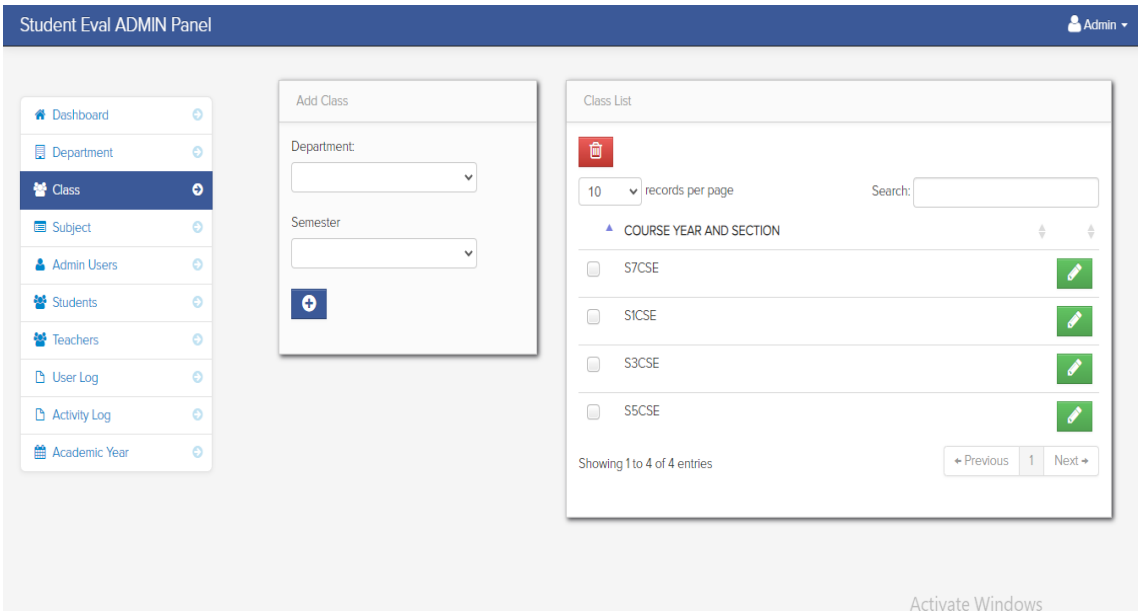


Figure 9.2: Add and view class

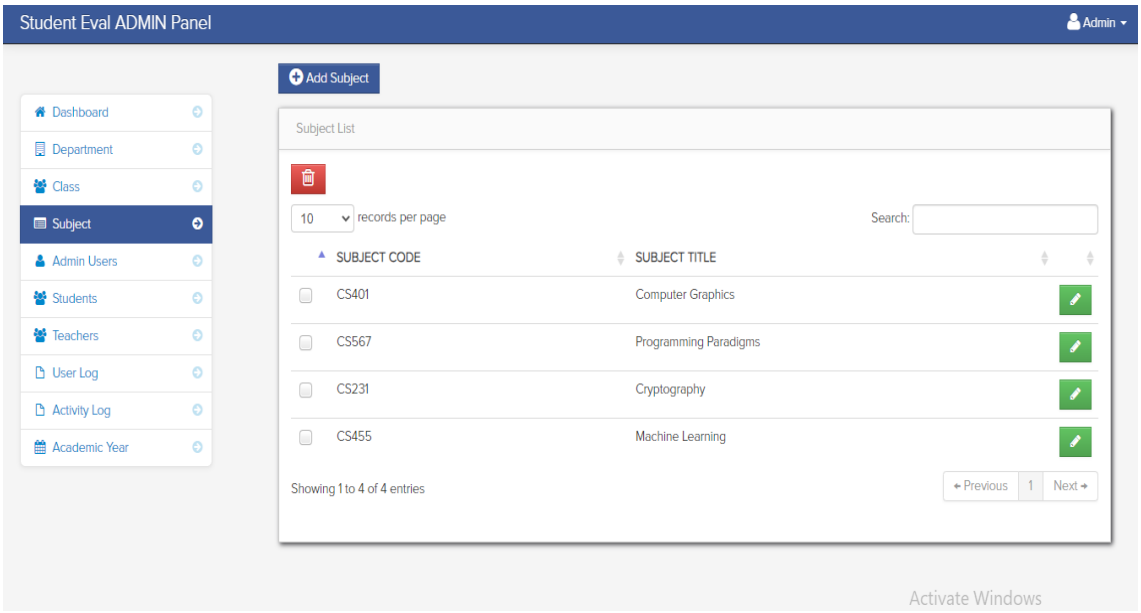


Figure 9.3: Add and view subjects

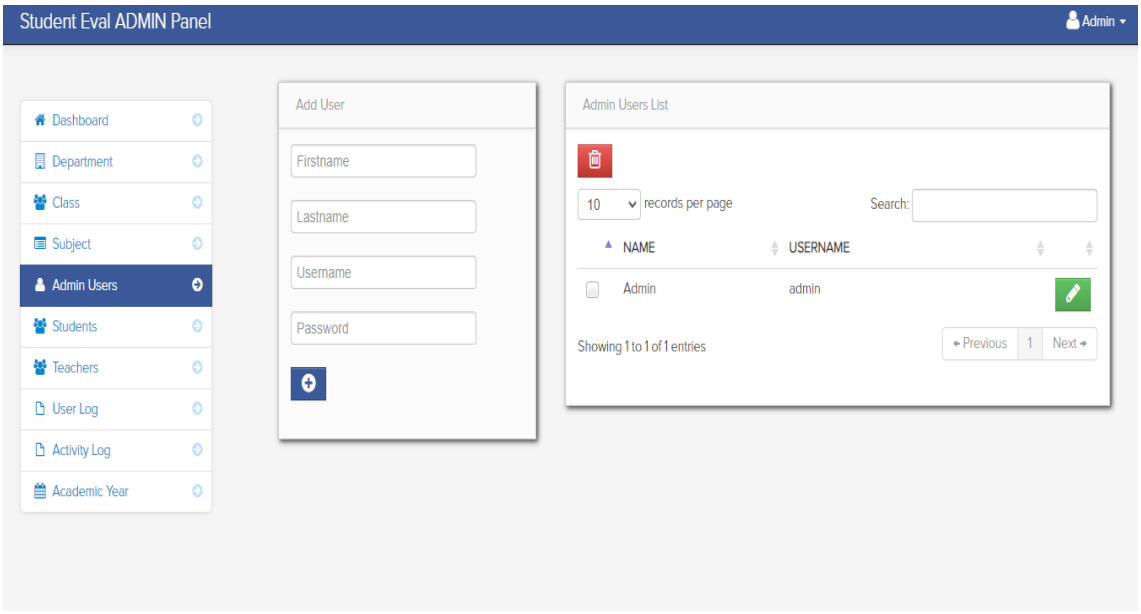


Figure 9.4: Admin users

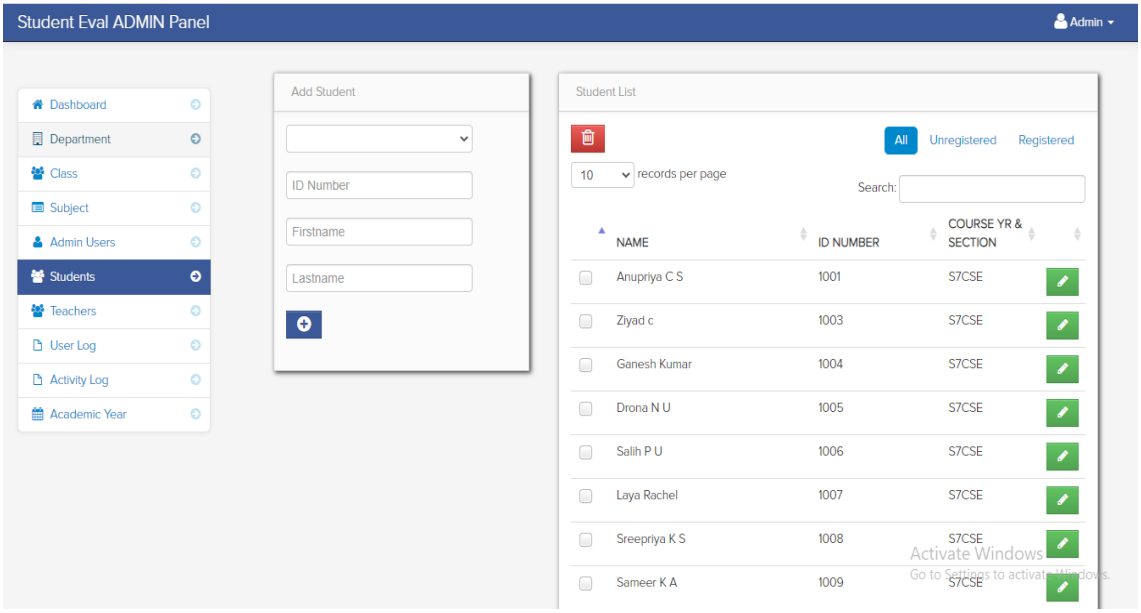


Figure 9.5: Adding students by admin

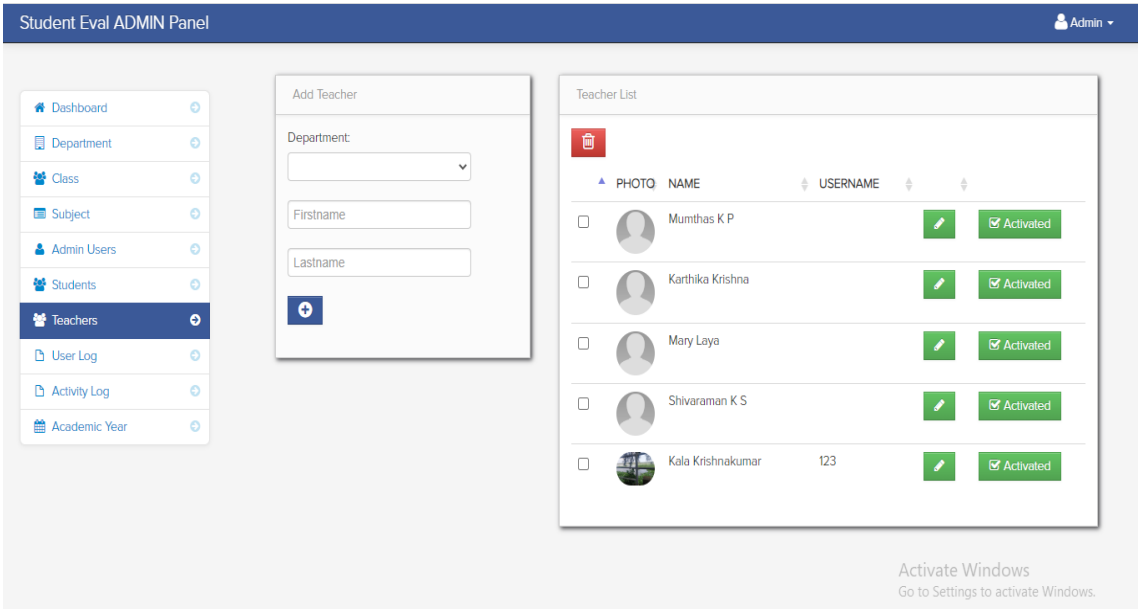


Figure 9.6: Adding Teacher by admin

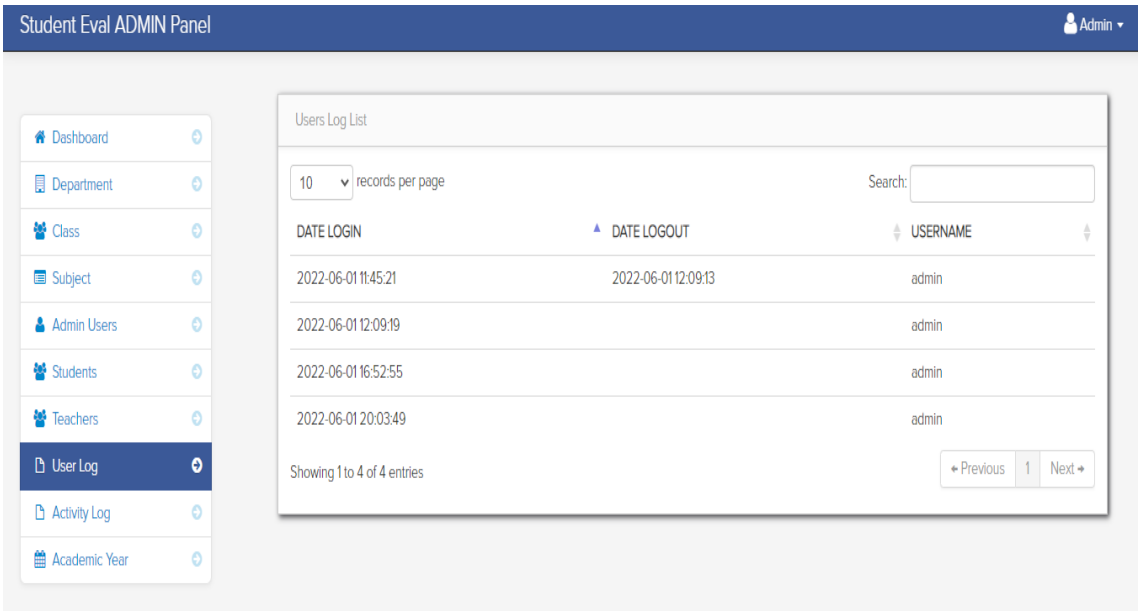


Figure 9.7: User log

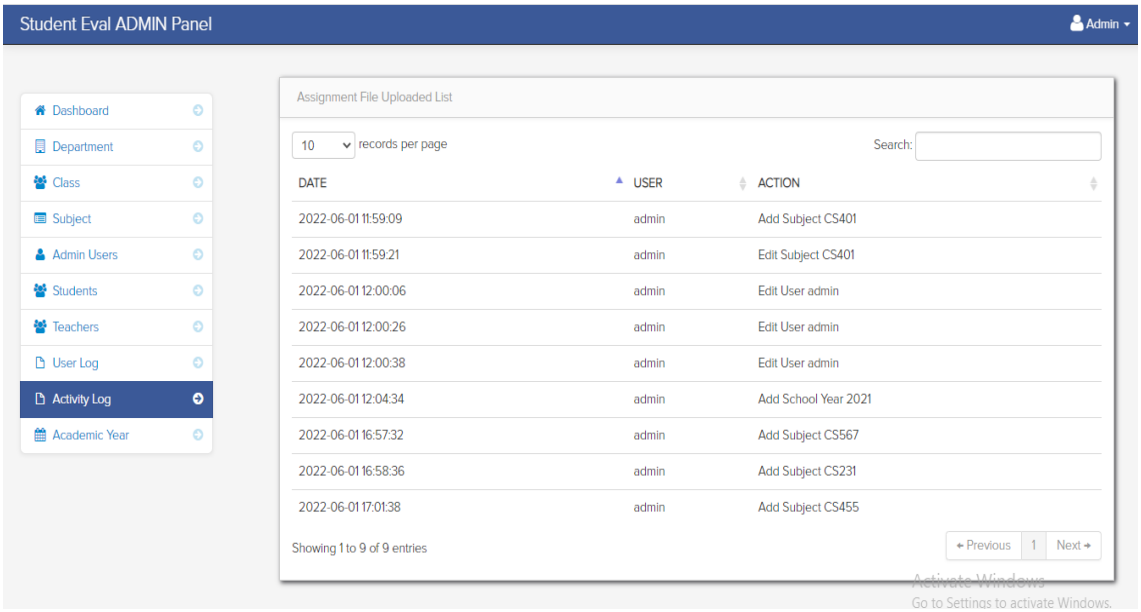


Figure 9.8: Activity log

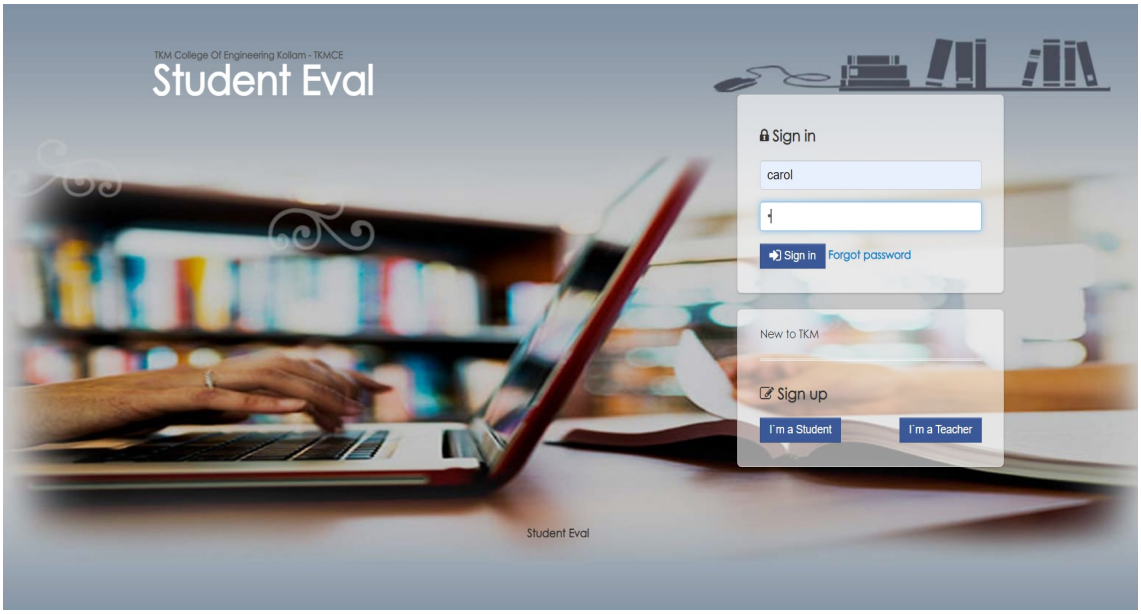


Figure 9.9: Teacher login

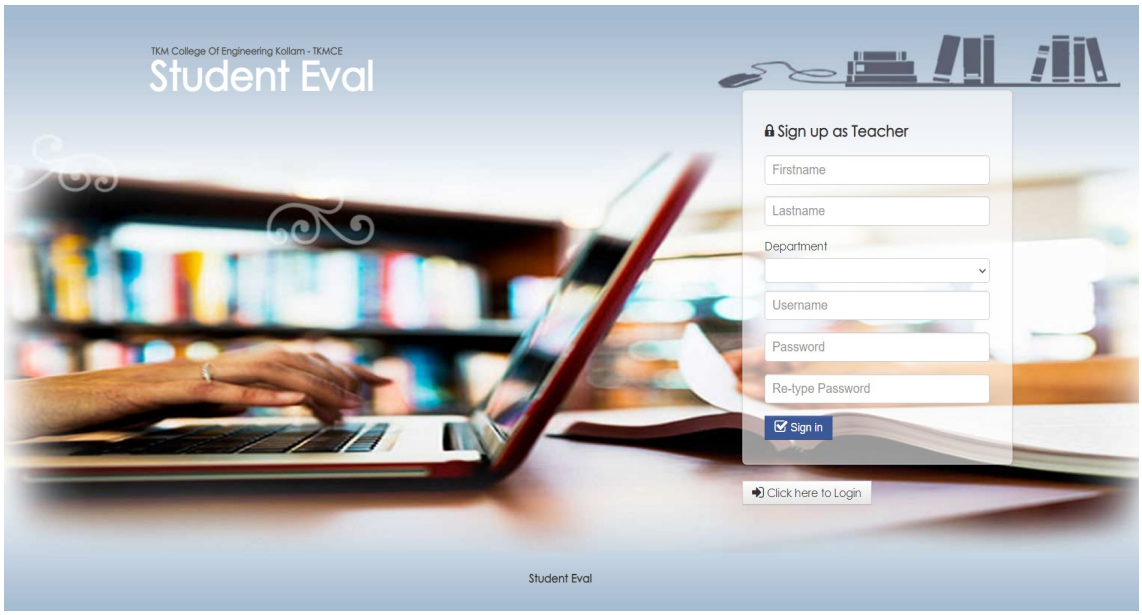


Figure 9.10: Teacher registration

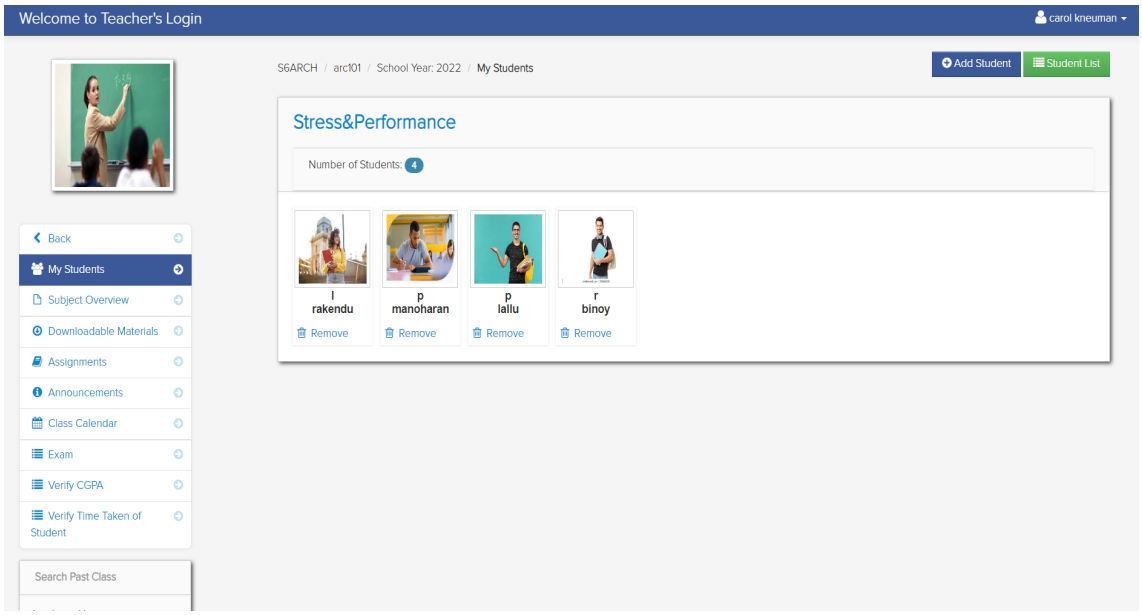


Figure 9.11: Students who are in the course

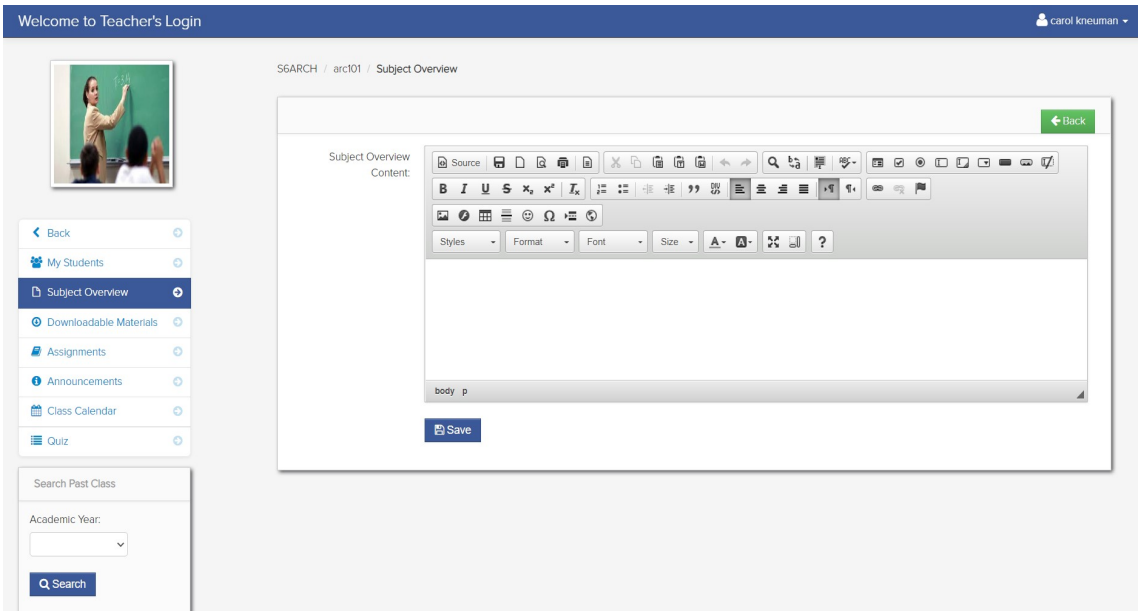


Figure 9.12: Subject overview

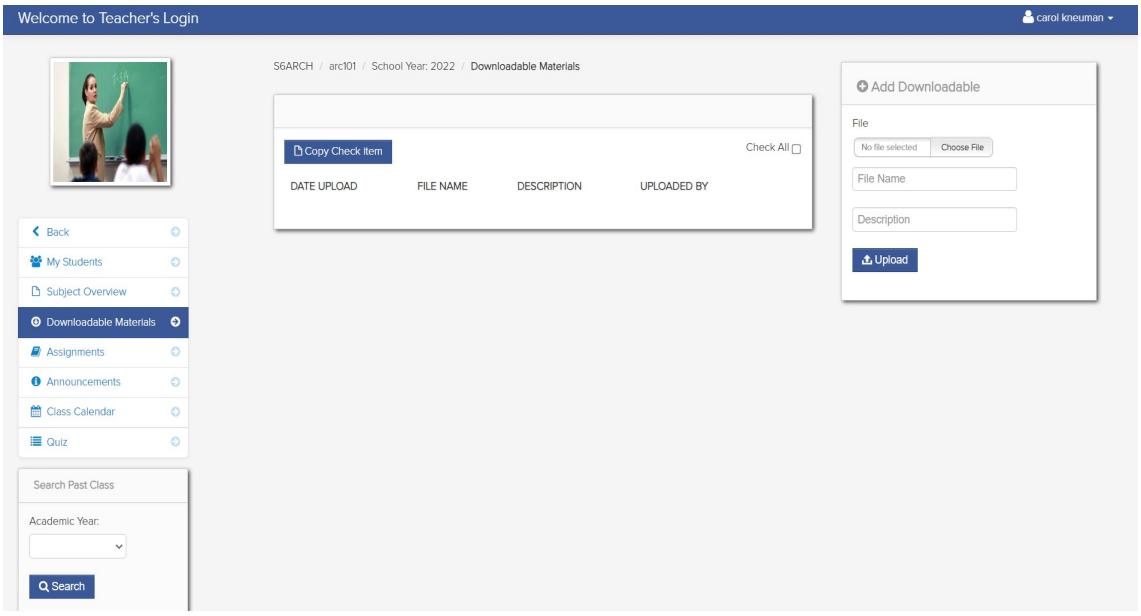


Figure 9.13: Downloadable material

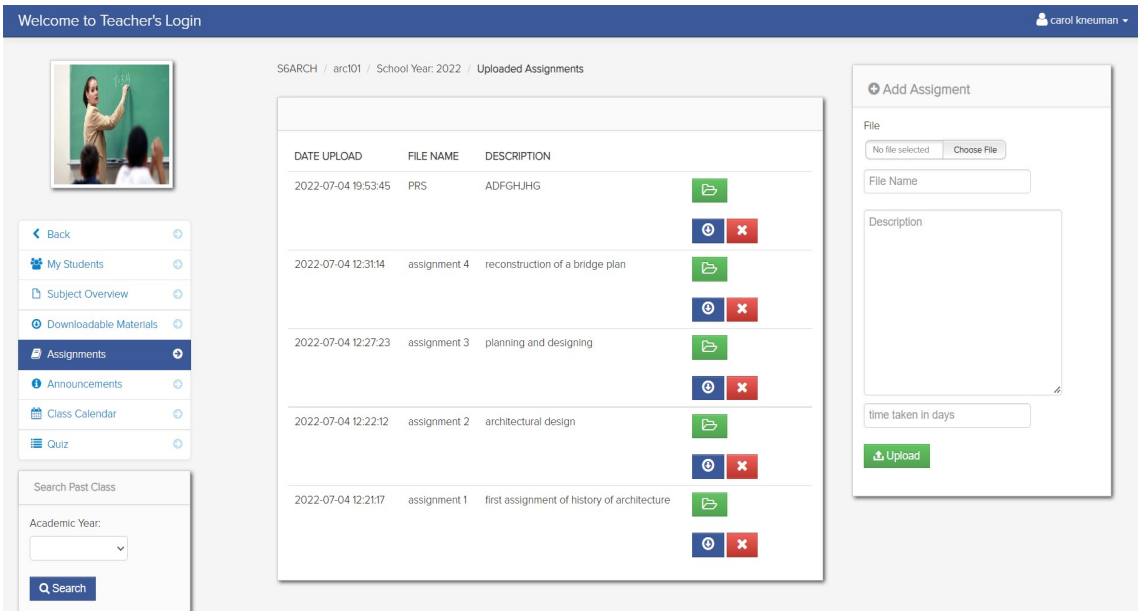


Figure 9.14: Adding assignment by teacher

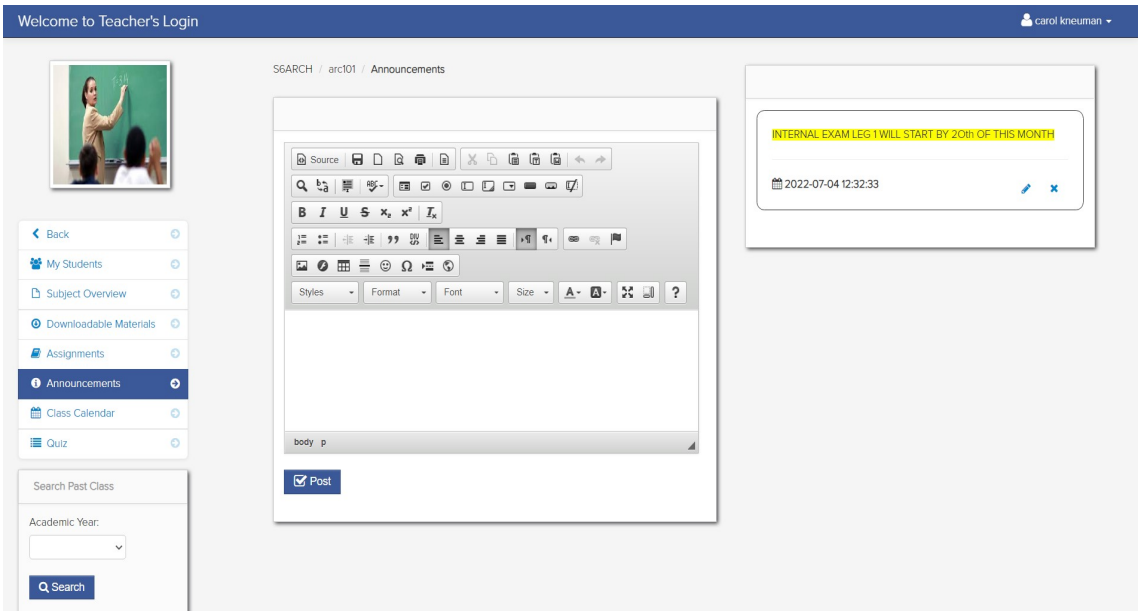


Figure 9.15: Announcement

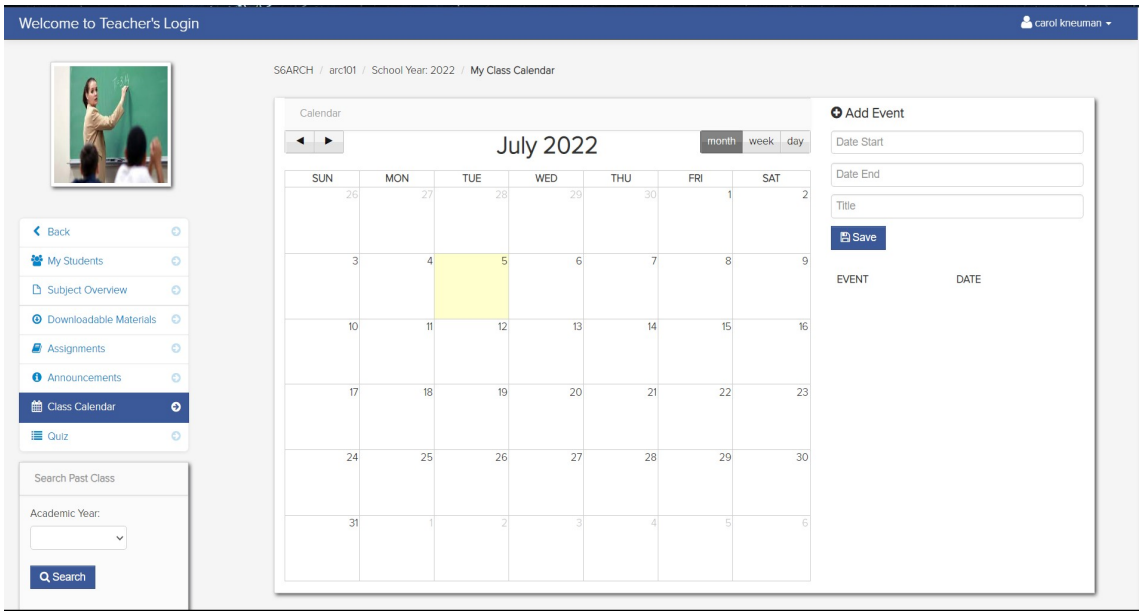


Figure 9.16: Class calendar

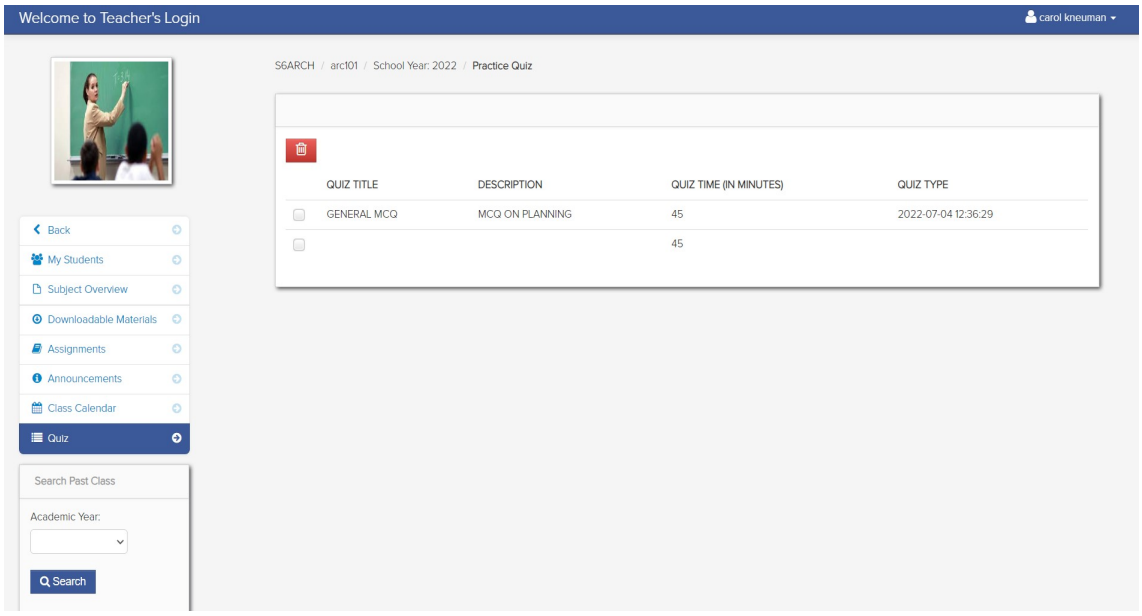


Figure 9.17: Quiz

Figure 9.18: Cgpa verification by teacher

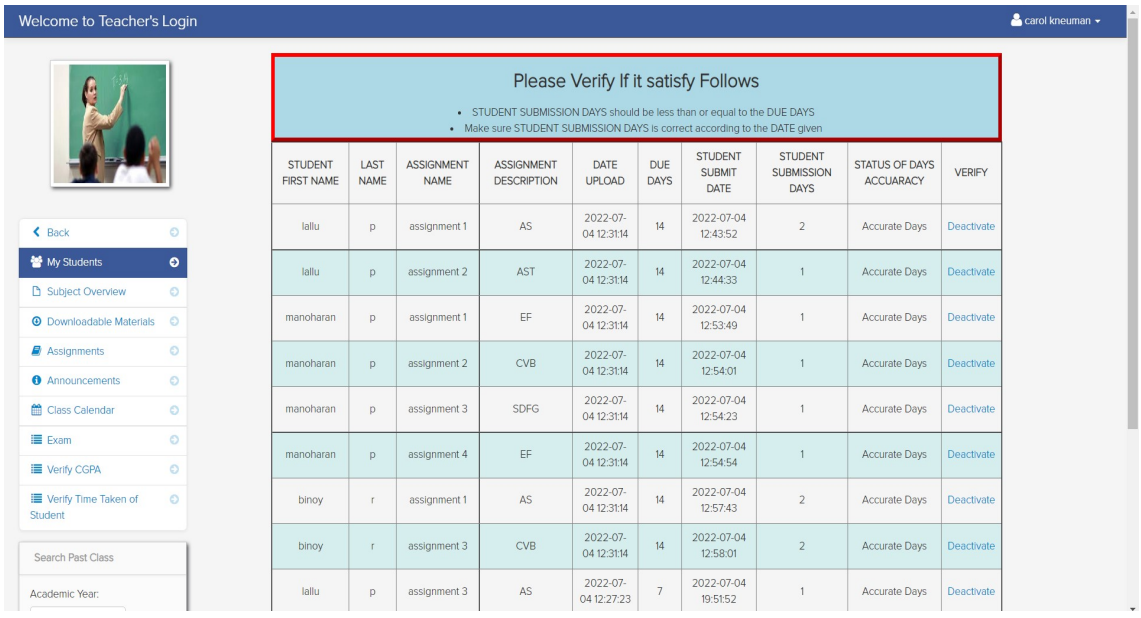


Figure 9.19: Assignment verification by teacher

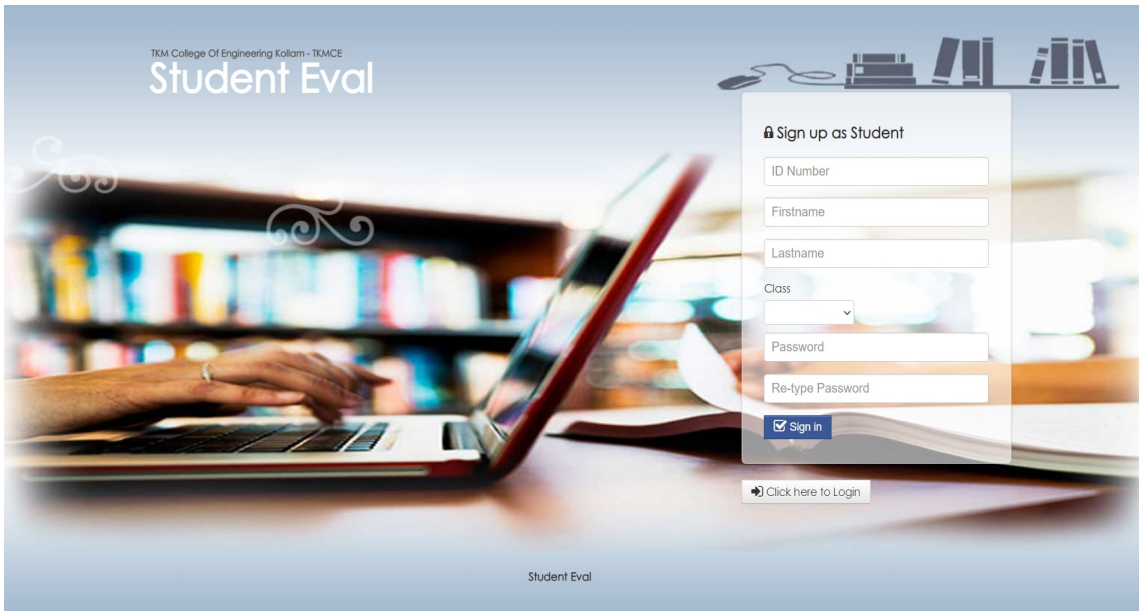


Figure 9.20: Student registration

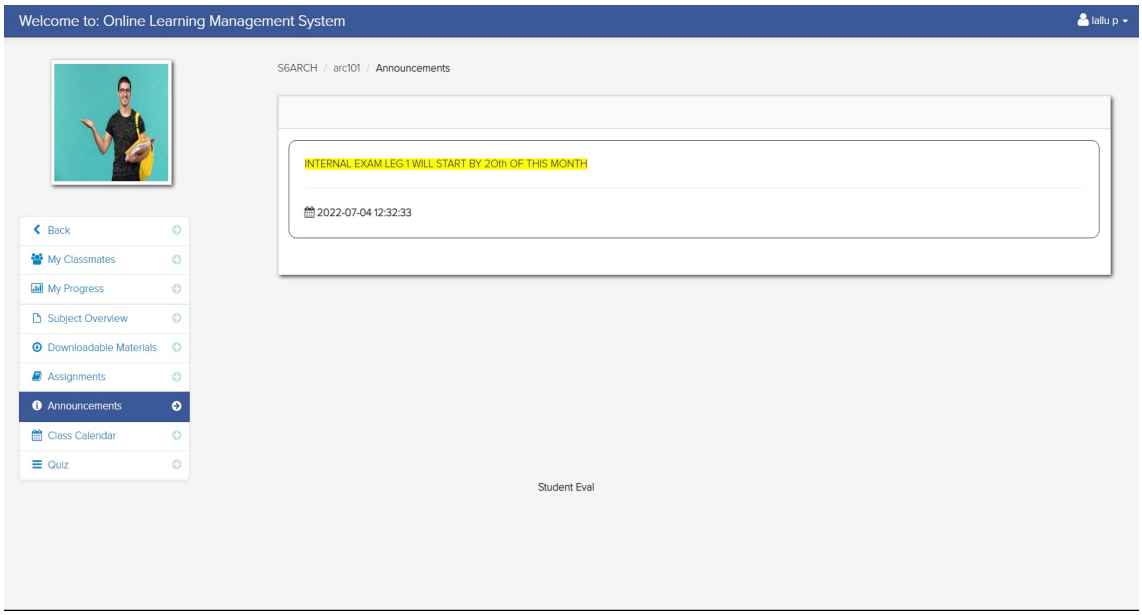


Figure 9.21: Student Dashboard

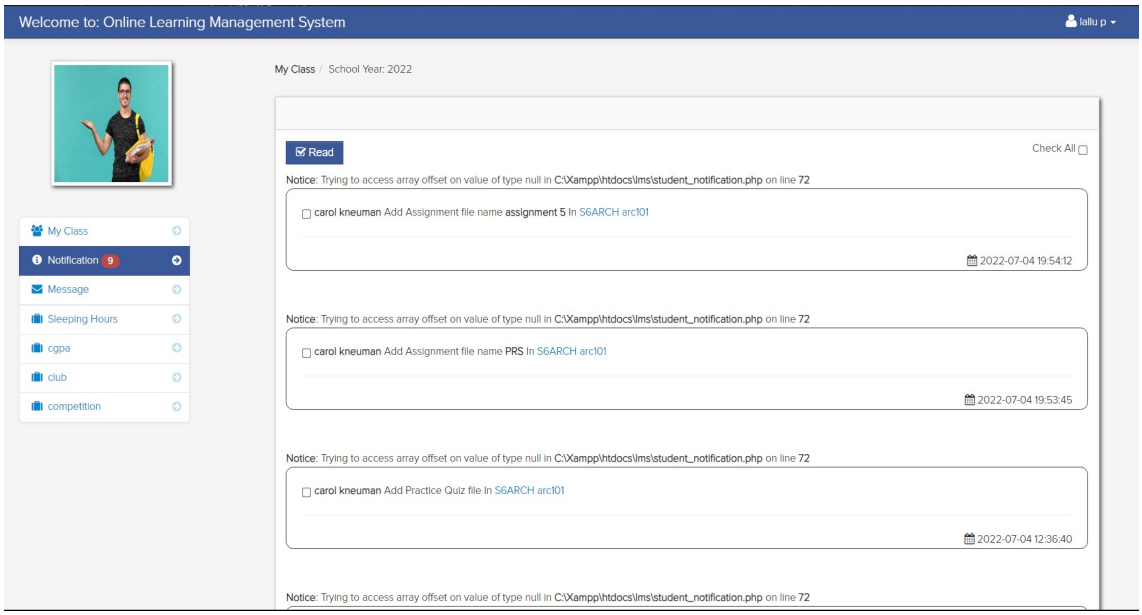


Figure 9.22: Notifications

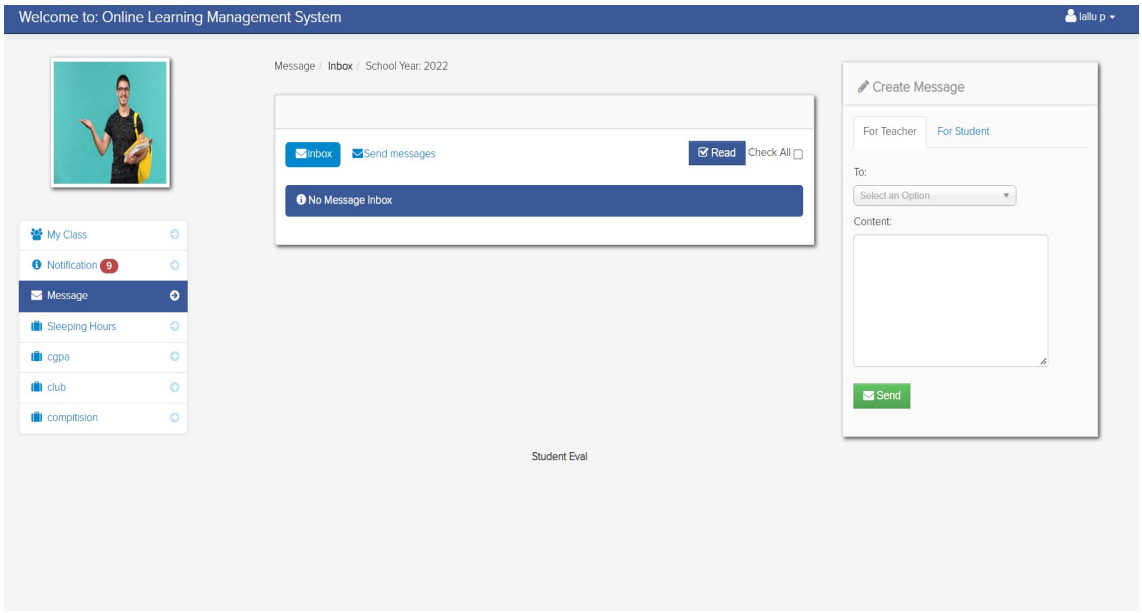


Figure 9.23: Message option for teacher and student

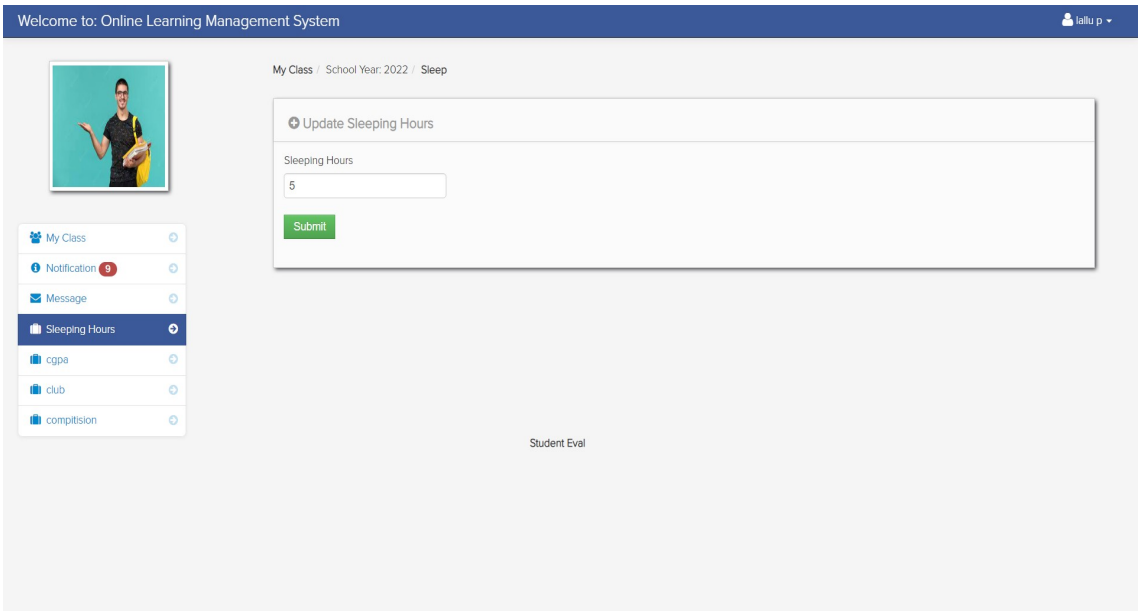


Figure 9.24: Sleeping hours

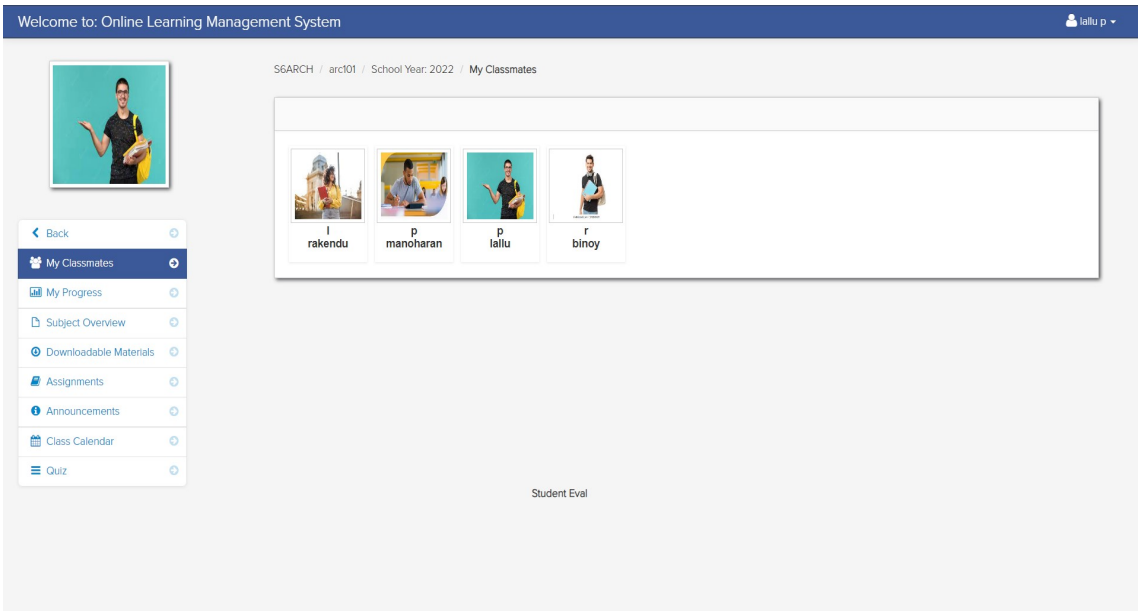


Figure 9.25: My Classmates

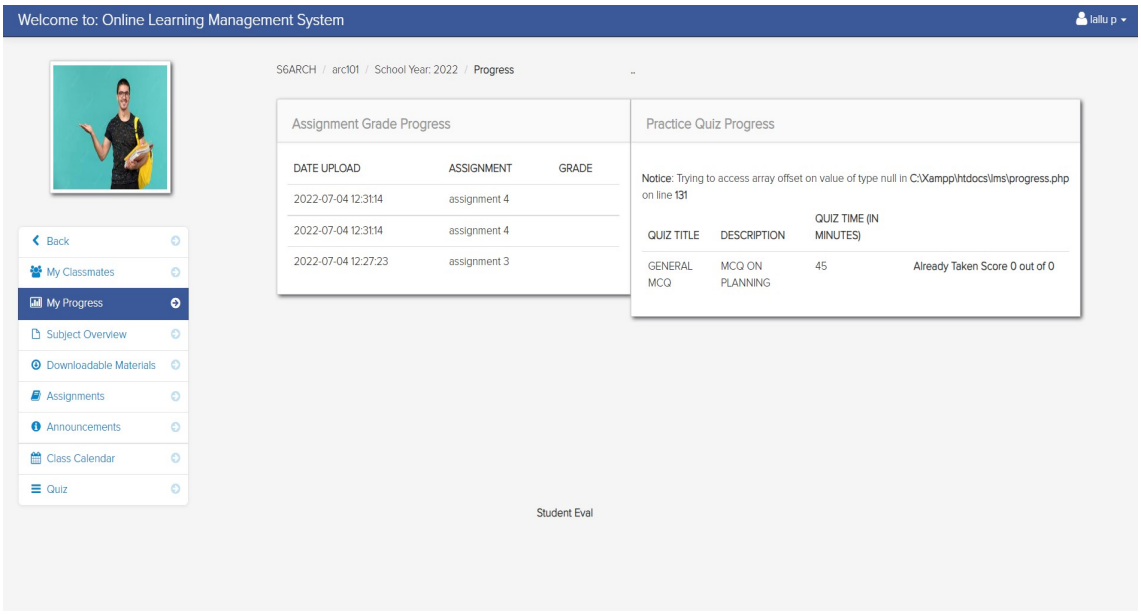


Figure 9.26: My Progress

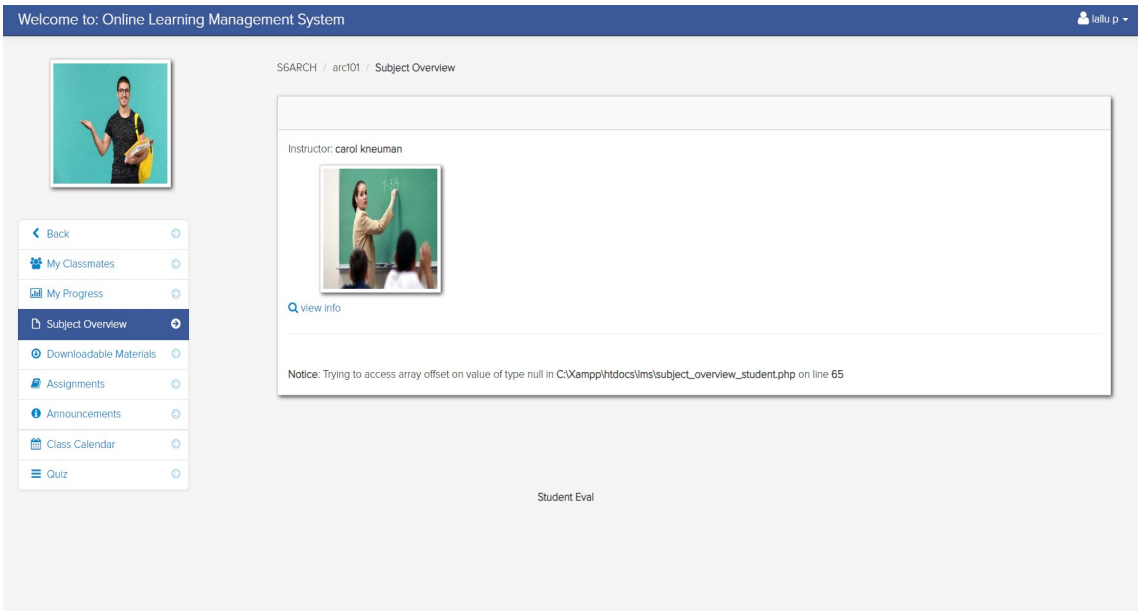


Figure 9.27: Subject Overview

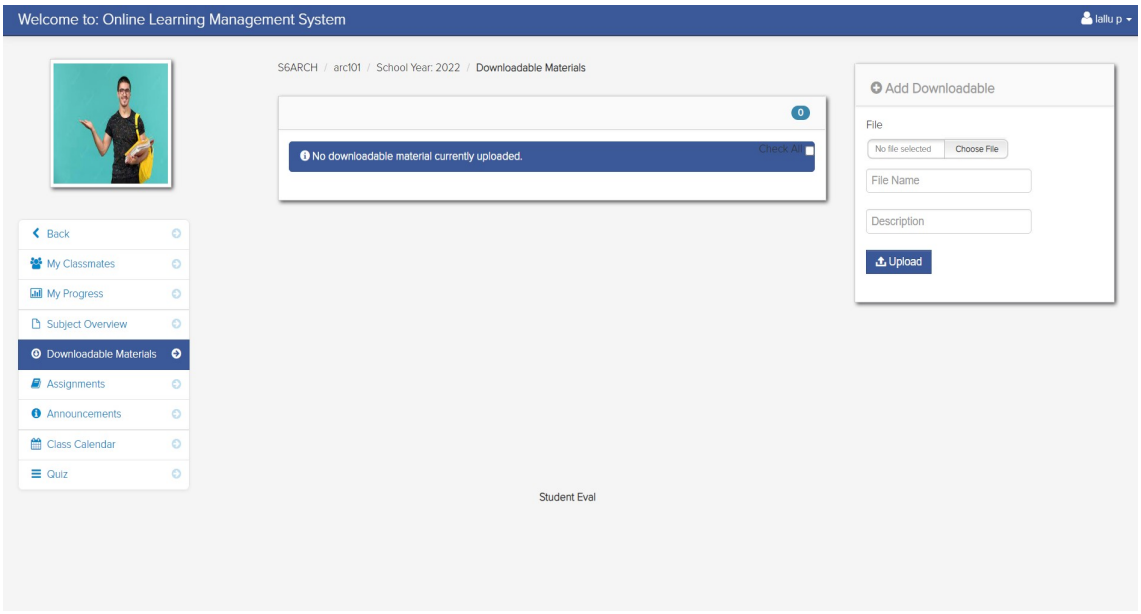


Figure 9.28: Downloadable Material

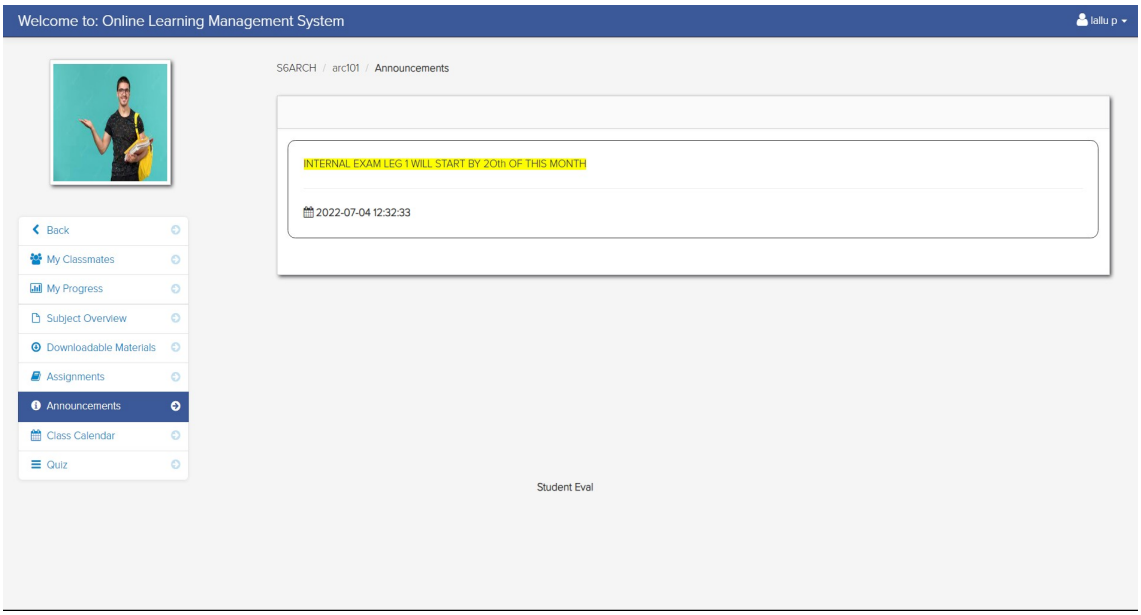


Figure 9.29: Announcements