**1 INTRODUCTION**

**1.1 INTRODUCTION**

The Performance Analysis using Sentiment Analyzer is a management information analysis system for educational institutes to manage student feedback data.

This system provides facilities for selecting any particular subject for feedback and generate a report automatically, build collective opinion of the students, student’s needs and requirements in the college. This system is an automatic feedback generation system that provides the proper feedback about the teachers by using categories like punctuality, explanation,use of black board, interactive, etc.

In the earlier existing system, students have to give the feedback on paper. After filling these paper feedback forms, these paper forms are collected by the faculties and class-teachers and then they calculate the overall grade for each subject and each teacher/faculty. Later on this collective grade report is viewed by the Head of Department. And then estimating the performance of teachers and course. This existing system is very time consuming and also causes a lot of trouble to the persons who are in charge of creating this analysis through the paper feedback forms. These are huge disadvantages of the existing system, thus the implementation of the Online Student Feedback Analysis System is very much necessary.

Student feedback on courses is also an essential element in quality assurance. Questionnaires are of huge importance in the dialogue with the students, since these questionnaires are the best tool currently available for collecting objective, detailed, and reasonably systematic information on a wide range of questions, which informs the teacher about student’s perception about the course – its strengths and weaknesses. Responses are gathered and analyzed on behalf of the department by this system and will be used for the purpose of the quality enhancement. The aim of this system is to save time and also to decrease human load and efforts.

**1.2 LITERATURE SURVEY**

**1.2.1 Introduction to machine learning:**

Analyzing the sentiments and accurate classification of this gigantic amount of data is a very challenging task. Most of the data available on the internet is in the textual form as it is the most natural and readable form for presenting the thoughts and opinions to the users . Machine Learning algorithms and techniques for sentiment analysis are deeply analyzed. These algorithms are more adaptive to changing inputs.

Machine learning techniques are generally used for binary classification and predictions of sentiments as either positive or negative.

Machine learning algorithms are further classified in the following categories as explained by

**1.2.1.1 Supervised:**

In these algorithms training dataset with the pre labeled classes are given and on the basis of this trained dataset the inputs are labeled with the output class/result.These algorithms classify the input data set with the help of trained classifier. Training data is composed of a set of training examples, each of them comprise of input object and desired output results. An inferred function is created by analyzing the training data by supervised learning methods that can be later used for mapping new incoming data which is also called the test data. Mostly machine learning techniques use the Supervised approach. It can be further categorized in two methodologies i.e., Classification and Regression. Most common examples of supervised machine learning algorithms are Linear Regression, Random Forest and Support Vector Machines.

**1.2.1.2 Un-supervised:**

These type of machine learning algorithms takes the unlabeled input data and then with the help of different algorithms hidden structure/pattern is discovered Unlike the supervised learning this technique does not use the pre labeled data to train the classifier. Un-supervised machine learning can be further divided into clustering and association, the most common example of Un-supervised machine learning algorithms are K-Means and Apriori Algorithm.

**1.2.1.3 Semi-Supervised:**

These type of algorithms deal with the both labeled and unlabeled data sets. Reviewed different lexicon based tools and techniques and mentioned the comparison between the features and accuracy results of different lexicon techniques. Taking it a step ahead, different Machine Learning techniques/algorithms are studied and analyzed in this research. A comprehensive analysis is also formulated between different techniques and accuracies.

**1.2.2 Introduction to sentiment analysis**

The sentiment analyzer is build using machine learning algorithms. There’s an algorithm which analyzes the descriptive type questions. The algorithm is trained using the training data set which contains positive and negative words. And then the model is created using the trained algorithm. Then the test data is given to the algorithm to see the accuracy of the system. If it lacks in accuracy, then more training data and features are added to the analyzer to increase the accuracy of the system and in such manner the machine learning is implemented to build the sentiment analyzer. The taking of feedback plays a very significant role in the life of students as well as the teachers.

Sentiment classification identifies sentiment polarity (positive or negative) from text (sentence or document), has been the most extensively studied task in sentiment analysis. Until now, there have been two mainstream approaches for sentiment classification. The lexicon-based approach aims to aggregate the sentiment polarity of a sentence from the polarity of words or phrases found in the sentence, and the learning-based approach treats sentiment polarity identification as a special text classification task and focuses on building classifiers from a set of sentences (or documents) annotated with their corresponding sentiment polarity. The lexicon-based sentiment classification approach is simple and interpretable, but suffers from scalability and is inevitably limited by sentiment lexicons that are commonly created manually by experts. It has been widely recognized that sentiment expressions are colloquial and evolve over time very frequently. . In practice, it is not feasible to create and maintain sentiment lexicons to capture sentiment expressions with high coverage. On the other hand, the learning-based approach relies on large annotated samples to overcome the vocabulary coverage and deals with variations of words in sentences. collect a large number of training corpora to train the sentiment classifier.

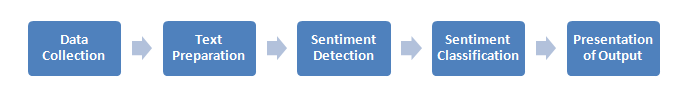


Figure 1.1 Sentiment Analysis Process

**1.2.2.1 Approach of Sentiment Analysis in the Proposed System**

In the proposed Performance analysis using sentiment analyzer, reviews given by the students about the course structure, contents of the course and faculties are recorded and then analyzed using sentiment analysis algorithms.

Firstly, we are using the General Sentiment Analysis Algorithm which is tuned to analyse the sentiment of a general statement/review. This General Sentiment Analysis Algorithm accepts any particular string and in return gives an emotion with the help of a rating for positive, negative and neutral. Then we have a compound result which provides the complete rating of the string.

**Input Example:**

{

“sentenceList“:[

“I like computer programming “,

“I love data mining and sentiment analysis”,

“I don’t want to learn physics.”

]

}

**Output Example:**

[

{

“positive “: 0.455,

“negative” :0,

“sentence”: I like computer programming “,

“neutral”:0.545,

},

{

“positive “: 0.673,

“negative “: 0,

“sentence”: I love data mining and sentiment analysis”,

“neutral”:0.488,

},

{

“positive “:0,

“negative”: 0.432,

“sentence”: “I hate physics”,

“neutral”: 0.668

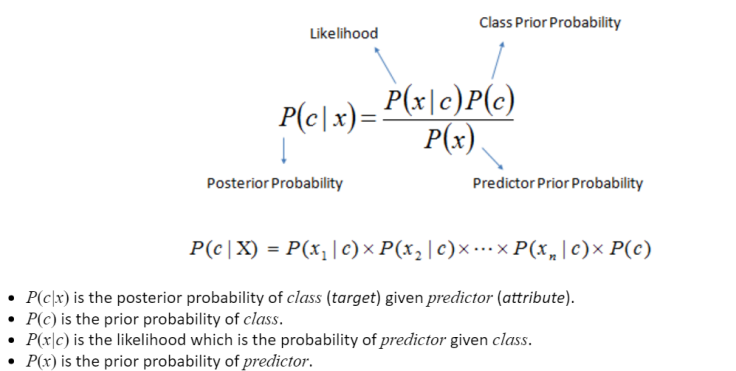
},

]

**1.2.3 Introduction to Navie Baye’s Classifier**

A Naive Bayes classifier is a probabilistic machine learning model that is used for classification of text. The Naive Bayesian classifier is based on Bayes’ theorem with the independent assumptions between predictors.

Bayes’ theorem relates the conditional and marginal probabilities of events c and x:



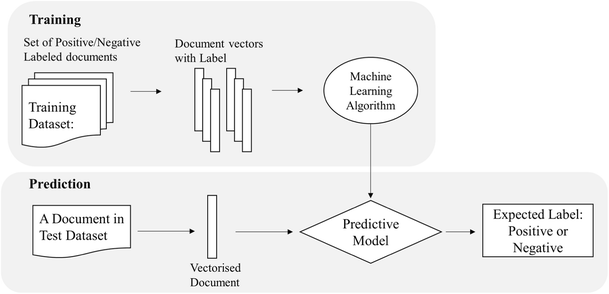


Figure 1.2 Naive Baye’s flowchart

**2. SYSTEM ANALYSIS**

**2.1 Existing System**

In the existing system students used to give feedbacks manually on paper feedback forms. It was a time consuming and very inefficient process. Then the forms are collected and the HOD’s views the feedbacks of students and analyze the performance of that teacher of that particular department. Then to overcome the limitations of that system came the online feedback systems which takes the feedback of students online and automatically analyzes the feedback to analyze the performance of teachers. But the existing online feedback systems only analyzes the objective type questions it doesn’t analyzes the descriptive type questions. In this project a sentiment analyzer is implemented to analyze the descriptive type questions so to increase the accuracy of the feedback system.

**2.1.1 Disadvantages of Existing System**

* It is time consuming job.
* It is difficult to maintain the feedback details.
* Preparation of reports is not an easy work.
* Manually maintaining the data’s is tedious and tedious and sometimes information may be lost or overloaded by human.

**2.2 PROPOSED SYSTEM**

The taking of feedback plays a very significant role in the life of students as well as the teachers. The students give the feedback so to convey what is the difference between the actual teaching which is currently taking place in colleges and what type of teaching students really desire for. And these feedbacks show the teachers their overall performance in their particular subjects. They can improve their teaching accordingly then This system is a secured system. The identity of the students giving feedback is not disclosed to anyone not even the admin. And a single student can give only a single feedback to a particular teacher. The accounts of students are created by the admin so no one other than the students can give the feedback. Sentiment analysis has received much attention from research and industry communities recently. In this feedback system, a database is created which contains negative and positive words.

**2.2.1 Features of proposed system**

* Giving feedback online saves time in comparison to manual process.
* Teacher’s performance can be increased using the analyzed feedback.
* Descriptive type questions can be analyzed by using the sentiment analyzer which was not possible in previous online feedback analysis system.
* The identity of the student is kept anonymous. False feedback can be prevented from submission.

**2.3 SYSTEM REQUIREMENTS**

**2.3.1 Hardware Requirements**

System : Intel core i3/i5/i7

RAM : 2 GB

Keyboard : 110 keys enhanced

**2.3.2 Software Requirements**

Operating system : Linux(Ubuntu)

Coding Language : Python

Database : Oracle 11g, MongoDB

Web Technologies : html, CSS, JavaScript

* 1. **System Modules**

In this system, there are basically three different modules which are as following:

- Admin Module

- Student Module

- Faculty Module

**2.4.1 Admin:-**

Firstly, there’s admin module which has admin login portal.Admin’s username and password is stored in database. The username and password of admin are checked with data present in the database. After login Admin creates the students as well as teachers account and fill their details. The whole data is stored in the database then. The admin also adds the subjects in the database. The admin also can delete the student’s as well as the teacher’s data. The admin can view all the feedbacks results present in the database.

**Functionalities of Admin:-**

* Admin Login
* Admin can add new faculty
* Admin can view faculty list
* Admin can add new student
* Admin can view student list
* Admin can assign subject to the faculty.
* Admin can check feedback given by the student

**2.4.2 Student:-**

Then there’s student module which has student login portal. Each and every valid student has their unique username and password which is given by admin. The username and password when entered are checked with data present in the database. After login student, in the feedback form the name of the teacher and the subject will automatically appears in the screen. After the submission of feedback the answers of all the questions are analyzed and the result is stored in database. If the student has already given the feedback of that particular teacher, then he can’t give the feedback again.

**Functionalities of Student:-**

* Student can login
* Student can view feedback status
* Student can give feedback
* Logout

**2.4.3 Faculty:-**

Then at last there’s teacher module which has teacher login portal. Each and every teacher has their unique username and password which is given by the admin. The username and password when entered are checked with data present in the database. The teachers can view their overall performance according to the student’s feedback. And student’s identity is not revealed to the teacher.

**Functionalities of Faculty:-**

* Faculty can login
* Faculty can check feedback given by the students
* Logout
  1. **FEASIBILITY STUDY**

After doing the project Performance Analysis using Sentiment Analyzer, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time.

**2.5.1** **ECONOMICAL FEASIBILITY**

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

**2.5.2 TECHNICAL FEASIBILITY**

This includes the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system and checked if everything was possible using different type of frontend and backend platforms.

* + 1. **OPERATIONAL FEASIBILITY**

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system.

**3. SYSTEM DESIGN**

**3.1 DATA FLOW DIAGRAM**

* The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
* It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

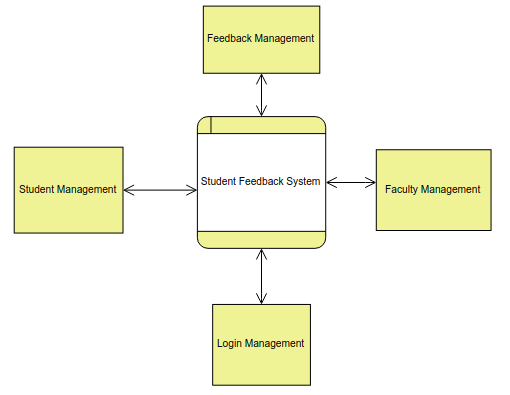


Figure 3.1 Data Flow Diagram Level 0

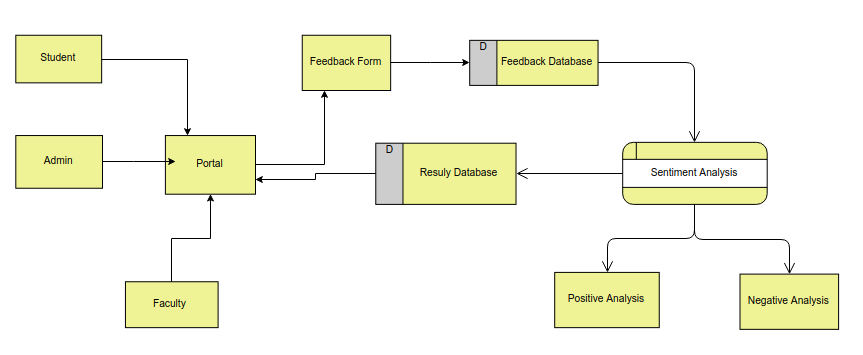


Figure 3.2 Data Flow Diagram Level 1

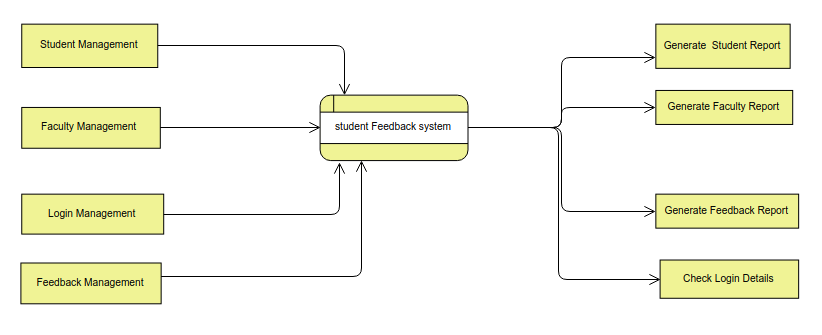


Figure 3.3 Data Flow Diagram Level 2

**3.2 UML**

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

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

Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

UML diagrams represent two different views of a system model:

* Static view
* Dynamic view

**Static (or *structural*) view:** emphasizes the static structure of the system usingobjects, attributes, operations and relationships. It includes class diagrams and composite structure diagrams.

**Dynamic (or *behavioral*) view:** emphasizes the dynamic behavior of the systemby showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

**3.2.1 USE CASE DIAGRAM**

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

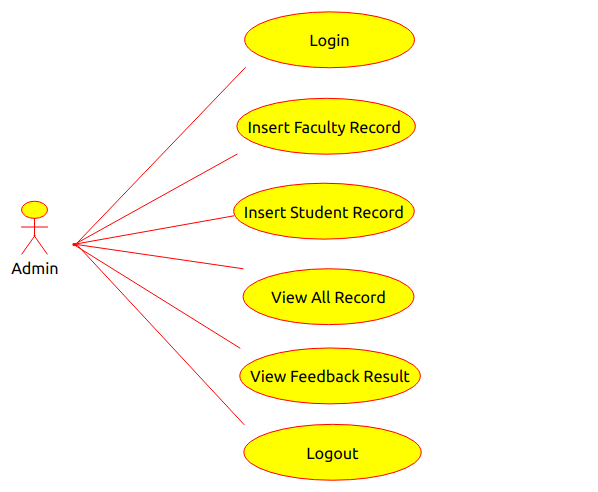


Figure 3.4 Use Case Diagram for Admin

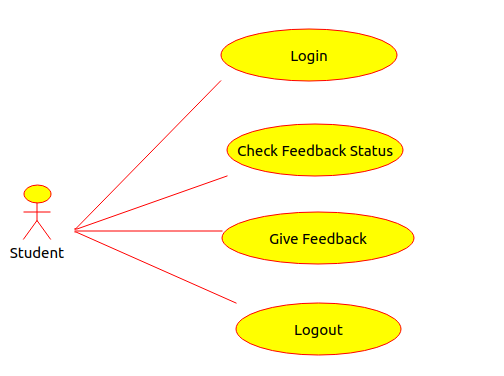


Figure 3.5 Use Case Diagram for Student



Figure 3.6 Use Case Diagram for Faculty

**3.2.2 SEQUENCE DIAGRAM:**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

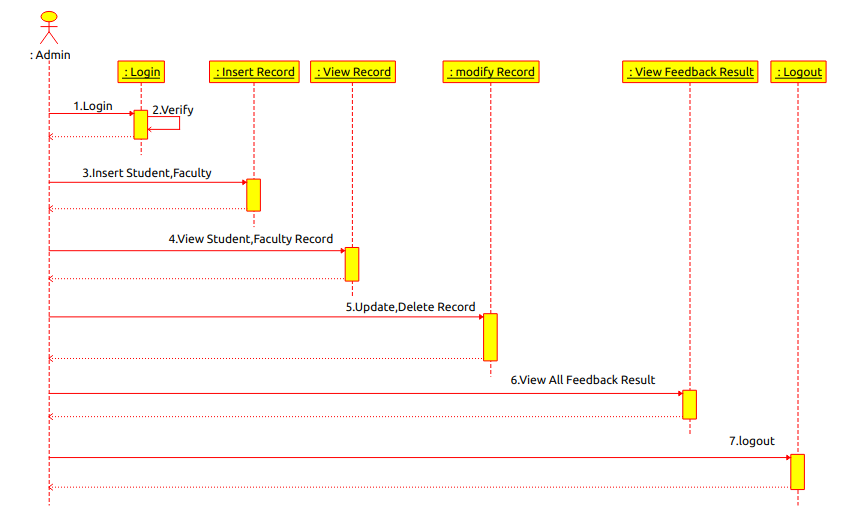


Figure 3.7 Sequence Diagram for Admin

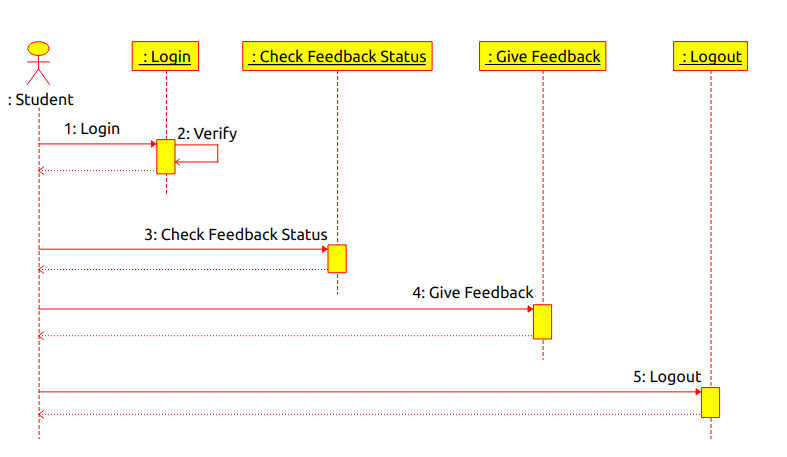


Figure 3.8 Sequence Diagram for Student

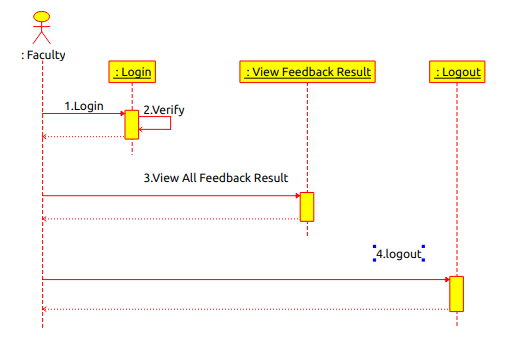


Figure 3.9 Squence Diagram for Faculty

**3.2.3 ACTIVITY DIAGRAM**

Activity diagrams describe the workflow behavior of a system. The diagrams describe the state of activities by showing the sequence of activities performed.

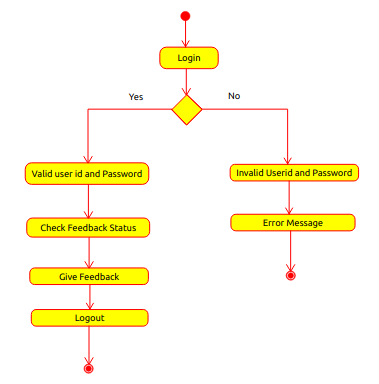


Figure 3.10 Activity Diagram for Student

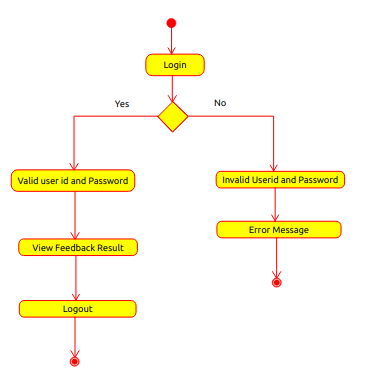


Figure 3.11 Activity Diagram for Faculty

**3.3 DATA DICTIONARY**

**3.3.1 FACULTY TABLE**

**Name Null? Type**

FID NOT NULL VARCHAR2(10)

FNAME VARCHAR2(40)

PASSWORD VARCHAR2(50)

EMAIL VARCHAR2(40)

PHONE NUMBER(38)

**3.3.2 STUDENT TABLE**

**Name Null? Type**

SID NOT NULL VARCHAR2(10)

SNAME VARCHAR2(50)

PASSWORD VARCHAR2(50)

BRANCH VARCHAR2(3)

EMAIL VARCHAR2(40)

PHONE NUMBER(38)

STATUS NUMBER(38)

FEEDBACK NUMBER(38)

**3.3.3 SUBJECT TABLE**

**Name Null? Type**

NAME NOT NULL VARCHAR2(50)

BRANCH VARCHAR2(5)

SEM VARCHAR2(5)

YEAR VARCHAR2(5)

**3.3.4 DETAILS TABLE**

**Name Null? Type**

FID VARCHAR2(10)

SUB NOT NULL VARCHAR2(40)

BRANCH VARCHAR2(3)

SEM VARCHAR2(3)

YEAR VARCHAR2(3)

**3.3.5 ADMIN TABLE**

Name Null? Type

USERNAME VARCHAR2(40)

PASSWORD VARCHAR2(50)

**4. SYSTEM OVERVIEW**

**4.1 Introduction to python**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

**4.2 Introduction to Anaconda Navigator**

**Anaconda**is a an open source distribution of the Python and R programming languages and it is used in data science, machine learning, deep learning-related applications  aiming at simplifying package management and deployment. Anaconda Distribution is used by over 7 million users, and it includes more than 300 data science packages suitable for Windows, Linux, and Mac OS.

Anaconda distribution comes with more than 1,400 packages as well as the [Conda](https://en.m.wikipedia.org/wiki/Conda_(package_manager)) package and virtual environment manager, called Anaconda Navigator  so it eliminates the need to learn to install each library independently.

The open source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. [Pip packages](https://en.m.wikipedia.org/wiki/Pip_(package_manager)) provide many of the features of conda packages and in most cases they can work together.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.m.wikipedia.org/wiki/Python_Package_Index) or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

**4.2.1 Anaconda Navigator**

Anaconda Navigator is a desktop [graphical user interface (GUI)](https://en.m.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.m.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOs and Linux.

The following applications are available by default in Navigator

* [JupyterLab](https://en.m.wikipedia.org/wiki/Project_Jupyter#Jupyter_Lab)
* [Jupyter Notebook](https://en.m.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* [QtConsole](https://qtconsole.readthedocs.io/en/latest/)
* [Spyder](https://en.m.wikipedia.org/wiki/Spyder_(software))
* [Glueviz](http://glueviz.org/)
* [Orange](https://en.m.wikipedia.org/wiki/Orange_(software))
* [Rstudio](https://en.m.wikipedia.org/wiki/Rstudio)
* [Visual Studio Code](https://en.m.wikipedia.org/wiki/Visual_Studio_Code)

**4.2.2 Conda**

Conda is an [open source](https://en.m.wikipedia.org/wiki/Open-source_software),cross platfoem, language-agnostic[package manager](https://en.m.wikipedia.org/wiki/Package_manager) and environment management system that installs, runs, and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., R), including multi-language projects. The Conda package and environment manager is included in all versions of Anaconda, Miniconda and Anaconda Repository.

**4.3 Jupyter Notebook**

Jupyter [Notebook](https://en.m.wikipedia.org/wiki/Notebook_interface) (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebooks documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a [JSON](https://en.m.wikipedia.org/wiki/JSON) document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.m.wikipedia.org/wiki/Markdown)), mathematics, plots and rich media, usually ending with the ".ipynb" extension.

A Jupyter Notebook can be converted to a number of [open standard](https://en.m.wikipedia.org/wiki/Open_standard) output formats ([HTML](https://en.m.wikipedia.org/wiki/HTML), [presentation slides](https://en.m.wikipedia.org/wiki/Presentation_slide), [LaTeX](https://en.m.wikipedia.org/wiki/LaTeX), [PDF](https://en.m.wikipedia.org/wiki/PDF), [ReStructuredText](https://en.m.wikipedia.org/wiki/ReStructuredText), [Markdown](https://en.m.wikipedia.org/wiki/Markdown),[Python](https://en.m.wikipedia.org/wiki/Python_(programming_language))) through "Download As" in the web interface, via the [nbconvert](https://nbconvert.readthedocs.io/) library or "jupyter nbconvert" command line interface in a shell.

To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through [NbViewer](https://nbviewer.org/) which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

**4.4 Oracle**

Oracle database is a relational database management system. It is known as Oracle database, OracleDB or simply Oracle. It is produced and marketed by Oracle Corporation.

Oracle database is the first database designed for enterprise grid computing. The enterprise grid computing provides the most flexible and cost effective way to manage information and applications.

Following are the four editions of the Oracle database.

* **Enterprise Edition:** It is the most robust and secure edition. It offers all features, including superior performance and security.
* **Standard Edition:**It provides the base functionality for users that do not require Enterprise Edition's robust package.
* **Express Edition (XE):** It is the lightweight, free and limited Windows and Linux edition.
* **Oracle Lite:** It is designed for mobile devices.

Oracle was originally developed by Lawrence Ellison (Larry Ellision) and his two friends and former co-worker in 1977. Oracle DB runs on the most major platforms like Windows, UNIX, Linux and Mac OS.

**5. SYSTEM IMPLEMENTATION**

**5.1 INTRODUCTION**

Implementation is the stage where the theoretical design is turned in to working system. The most crucial stage is achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding.

This project needs a open-source Anaconda Distribution. Project is implemented in python, so it can be run in any OS.

**5.2 SAMPLE CODE**

**HOME**

**HTML Code:**

<html>

<head>

<script>

$(document).ready(function(){

$("#login").click(function(){

$.ajax({

url: '/validation',

method: "POST",

data: $('form').serialize(),

success: function(response) {

if(response=="Success")

{

alert("Successfully Logged in");

window.location.href = $('#typ').val();

}

else

{

alert(response);

}

},

error: function(error) {

alert(error);

}

});

});

});

</script>

</head>

<body>

<div class="container">

<form onsubmit="return false;">

<table align="center" cellspacing="0px" cellpadding="20px">

<tr colspan=2><td><select name="type" id="typ">

<option>Admin</option>

<option>Student</option>

<option>Faculty</option>

</select></td></tr>

<tr><td><b>Username</b></td>

<td><input type="text" placeholder="Enter Username" name="username" id="uname" required></td></tr>

<tr><td><b>Password</b></td>

<td><input type="password" placeholder="Enter Password" name="password" id="pwd" required></td></tr>

<tr colspan=2><td><button type="submit" id="login">Login</button></td></tr>

</table>

</form>

</div>

</div>

</body>

</html>

**Python Code:**

@app.route('/home')

def home():

return render\_template('home.html')

**ADMIN**

**ADDFACULTY**

**HTML code**

<html>

<body>

<div style="width:500px;text-align:center;margin-top:25px;margin-left:100px;;margin-right:50px;">

<form id="form1" onsubmit="return false;">

<div class="form-group row">

<label for="fid" class="col-sm-2 col-form-label">Id</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="fid" name="fid" required="true">

</div>

</div>

<div class="form-group row">

<label for="fname" class="col-sm-2 col-form-label">Name</label>

<div class="col-sm-10">

<input type="text" class="form-control" name="fname" id="fname" required="true">

</div>

</div>

<div class="form-group row">

<label for="Email" class="col-sm-2 col-form-label">Email</label>

<div class="col-sm-10">

<input type="email" class="form-control" name="Email" id="Email" required="true">

</div>

</div>

<div class="form-group row">

<label for="phone" class="col-sm-2 col-form-label">Phone</label>

<div class="col-sm-10">

<input type="number" class="form-control" name="phone" id="phone" required="true">

</div>

</div>

<button type="submit" id="addf">Submit</button>

</form>

<script>

$(document).ready(function(){

$("#addf").click(function(){

$.ajax({

url: '/fdetails',

method: "POST",

data: $('#form1').serialize(),

success: function(response) {

alert(response);

$("#form1").reset();

},

error: function(error) {

alert("error");

}

});

});

});

</script>

</div>

</body>

</html>

**Python code**

@app.route('/fdetails',methods=['GET','POST'])

def fdetails():

fid=request.form['fid']

fname=request.form['fname']

emil=request.form['Email']

mob=request.form['phone']

conn.execute('select fid from faculty where fid='+fid)

row=conn.fetchone()

if row is None:

sqlStr = "INSERT INTO faculty VALUES(:id,:name,:pwd,:email,:mob)"

conn.execute (sqlStr, {'id':fid,'name':fname,'pwd':fid,'email':emil,'mob':mob})

conn.execute ('commit')

return "Success"

else:

return "faculty details already exists"

**ADDSTUDENT**

**HTML Code**

<html>

<body>

<div style="width:500px;text-align:center;margin-top:25px;margin-left:100px;;margin-right:50px;">

<form onsubmit="return false;">

<div class="form-group row">

<label for="sid" class="col-sm-2 col-form-label">Id</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="sid" name="sid" required="true">

</div>

</div>

<div class="form-group row">

<label for="sname" class="col-sm-2 col-form-label">Name</label>

<div class="col-sm-10">

<input type="text" class="form-control" name="sname" id="sname" required="true">

</div>

</div>

<div class="form-group row">

<label for="branch" class="col-sm-2 col-form-label">Branch</label>

<div class="col-sm-10">

<select class="form-control" name="branch" id="branch">

<option>CSE</option>

<option>ECE</option>

<option>EEE</option>

<option>MECH</option>

<option>CIVIL</option>

</select>

</div>

</div>

<div class="form-group row">

<label for="Email" class="col-sm-2 col-form-label">Email</label>

<div class="col-sm-10">

<input type="email" class="form-control" name="Email" id="Email" required="true">

</div>

</div>

<div class="form-group row">

<label for="phone" class="col-sm-2 col-form-label">Phone</label>

<div class="col-sm-10">

<input type="number" class="form-control" name="phone" id="phone" required="true">

</div>

</div>

<input type="submit" id="adds">

</form>

<script>

$("#adds").click(function(){

$.ajax({

type: "POST",

url: '/sdetails',

data: $('form').serialize(),

success: function(response) {

alert(response);

},

error: function(error) {

alert(error);

}

});

});

</script>

</div>

</body>

</html>

**Python Code**

@app.route('/sdetails',methods=['GET','POST'])

def sdetails():

sid=request.form['sid']

sname=request.form['sname']

branch=request.form['branch']

emil=request.form['Email']

mob=request.form['phone']

conn.execute('select sid from student where sid=\''+sid+'\'')

row=conn.fetchone()

if row is None:

sqlStr = "INSERT INTO student VALUES(:id,:name,:pwd,:branch,:email,:mob,:stat,:feed)"

conn.execute (sqlStr, {'id':sid,'name':sname,'pwd':sid,'branch':branch,'email':emil,'mob':mob,'stat':0,'feed':0})

conn.execute ('commit')

return "Successfully Inserted"

else:

return "Student details already exists"

**STUDENT**

**GIVEFEEDBACK**

**HTML Code**

<html>

<body>

{% for row in data %}

<div class="panel panel-default" style="margin-bottom:100px;">

<div class="panel-heading" style="background;rgb(230, 230, 250);"><h2>{{row[1]}}</h2><h5 style="margin-left:100px;"><b>- {{row[0]}}</b></h5></div>

<div class="panel-body">

<div class="row" style="margin-bottom:20px;margin-right:5px;">

<div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">

<div class="hexagon1" style="text-align:center;">

<img src="{{ url\_for('static',filename='student/interact.png')}}" style="margin-top:15px;margin-left:25px;" width="100px" height="80px">

<h4 style="padding-top:10px;text-align:center;">Explanation of Topic</h4>

<span class="bottom1"></span>

</div>

</div>

<div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">

<div class="hexagon1">

<img src="{{ url\_for('static',filename='student/pun.png')}}" style="margin-top:15px;margin-left:25px;" width="100px" height="80px">

<h4 style="padding-top:10px;text-align:center;">Punctuality</h4>

<span class="bottom1"></span>

</div>

</div>

<div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">

<div class="hexagon1">

<img src="{{ url\_for('static',filename='student/lab.png')}}" style="margin-top:15px;margin-left:25px;" width="100px" height="80px">

<h4 style="padding-top:10px;text-align:center;">Use of I.C.T</h4>

<span class="bottom1"></span>

</div>

</div>

<div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">

<div class="hexagon1">

<img src="{{ url\_for('static',filename='student/del.png')}}" style="margin-top:15px;margin-left:25px;" width="100px" height="80px">

<h4 style="padding-top:10px;text-align:center;">Clarity &amp Audibility</h4>

<span class="bottom1"></span>

</div>

</div>

<div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">

<div class="hexagon1">

<img src="{{ url\_for('static',filename='student/del.png')}}" style="margin-top:15px;margin-left:25px;" width="100px" height="80px">

<h4 style="padding-top:10px;text-align:center;">Use of Black Board</h4>

<span class="bottom1"></span>

</div>

</div>

<div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">

<div class="hexagon1">

<img src="{{ url\_for('static',filename='student/del.png')}}" style="margin-top:15px;margin-left:25px;" width="100px" height="80px">

<h4 style="padding-top:10px;text-align:center;">Seminars/ Projects</h4>

<span class="bottom1"></span>

</div>

</div>

</div>

<form id="{{row[1]}}" name="{{row[0]}}" style="float:left;">

<select style="max-width:200px;margin-left:20px;" id="Explanation of Topic"><option>Excellent</option><option >good</option><option>Avg</option><option>Poor</option></select>

<select style="max-width:200px;margin-left:120px;" id="Punctuality"><option>Punctual and Regular</option><option>Punctual but Irregular</option><option>Regular but not Punctual</option><option>Not Punctual and Irregular</option></select>

<select style="max-width:200px;margin-left:120px;" id="Use of I.C.T"><option>Excellent</option><option>good</option><option>Avg</option><option>Poor</option></select>

<select style="max-width:200px;margin-left:100px;" id="Clarity & Audibility"><option>Clear and Audible</option><option>Audible but Discontinuous</option><option>Spends Time with Irrelevant Talk</option><option>Neither Clear nor Audible</option></select>

<select style="max-width:200px;margin-left:70px;" id="Use of Black Board"><option>Excellent</option><option>good</option><option>Avg</option><option>Poor</option></select>

<select style="max-width:200px;margin-left:120px;" id="Seminars/ Projects"><option>Organizing</option><option>Encouraging</option><option>No</option></select>

</form>

</div>

</div>

{% endfor %}

<input type="submit" id="fedb">

<script>

$("#fedb").click(function(){

var result={};

$('form').each(function() {

var fname=$(this).attr('id');

var sname=$(this).attr('name');

var name={};

name[fname]={};

var entries=name[fname];

entries[sname]={};

var val1=entries[sname];

$(this).children('select').each(function() {

attri=$(this).attr('id');

val=$(this).val();

val1[attri]=val;

});

$.extend(result,name);

});

$.ajax({

type: 'POST',

contentType: 'application/json',

data: JSON.stringify(result),

dataType: 'json',

url: '/stufeed',

success: function (response) {

alert(response["success"]);

var lin='Student'

window.location.href=lin;

},

error: function(error) {

alert("error");

}

});

});

</script>

</body>

</html>

**Python Code:**

import pymongo

@app.route('/stufeed',methods=['POST'])

def stufeed():

if request.method == 'POST':

asd = request.json

user=session['user']

for fname in asd:

for sub in asd[fname]:

for k,v in asd[fname][sub].items():

if fname in mydb.list\_collection\_names():

col=mydb[fname]

col.update\_one({'\_id':fname},{'$push':{sub:{k:v}}})

else:

col=mydb[fname]

col.insert\_one({"\_id":fname,sub:[{k:v}]})

conn.execute('update student set feedback=1 where sid=\''+user+'\'')

conn.execute ('commit')

print("added success")

return jsonify({"success":"Feedback Successfully Submitted"})

**FACULTY**

**VIEWFEEDBACK**

**HTML Code**

<html>

<body>

<h1>Overall Performence of a Faculty</h1>

<p >{{ data|safe }}</p>

<div style="width:800px !important;height:500px !important;">

<canvas id="myChart" widths="10" height="50"></canvas>

</div>

<script>

var d = {{ data|safe }};

var ctx = document.getElementById("myChart").getContext("2d");

var total=d["Explanation of Topic"]["pos"]+d["Explanation of Topic"]["neu"]+d["Explanation of Topic"]["neg"];

var data = {

labels: ["Explanation of Topic","Punctuality","Use of I.C.T","Clarity & Audibility","Use of Black Board","Seminars/ Projects"],

datasets: [

{

label: "Positive",

backgroundColor: "green",

data: [(d["Explanation of Topic"]["pos"]/total)\*100,(d["Punctuality"]["pos"]/total)\*100,(d["Use of I.C.T"]["pos"]/total)\*100,(d["Clarity & Audibility"]["pos"]/total)\*100,(d["Use of Black Board"]["pos"]/total)\*100,(d["Seminars/ Projects"]["pos"]/total)\*100

]

}, {

label: "Negative",

backgroundColor: "red",

data: [(d["Explanation of Topic"]["neg"]/total)\*100,(d["Punctuality"]["neg"]/total)\*100,(d["Use of I.C.T"]["neg"]/total)\*100,(d["Clarity & Audibility"]["neg"]/total)\*100,(d["Use of Black Board"]["neg"]/total)\*100,(d["Seminars/ Projects"]["neg"]/total)\*100

]

}, {

label: "Neutral",

backgroundColor: "blue",

data: [(d["Explanation of Topic"]["neu"]/total)\*100,(d["Punctuality"]["neu"]/total)\*100,(d["Use of I.C.T"]["neu"]/total)\*100,(d["Clarity & Audibility"]["neu"]/total)\*100,(d["Use of Black Board"]["neu"]/total)\*100,(d["Seminars/ Projects"]["neu"]/total)\*100

]

}

]

};

var myBarChart = new Chart(ctx, {

type: 'bar',

data: data,

options: {

responsive: true,

maintainAspectRatio: false,

barValueSpacing: 5,

scales: {

yAxes: [{

ticks: {

beginAtZero:true,

Max: 100,

callback: function(value) {

return value + "%"

}

},

scaleLabel: {

display: true,

labelString: "Percentage"

}

}]

}

}

});

</script>

</body>

</html>

**Python Code**

import nltk.classify.util

import pdfkit

from nltk.classify import NaiveBayesClassifier

from nltk.corpus import names

def word\_feats(words):

return dict([(word, True) for word in words])

myclient=pymongo.MongoClient("mongodb://localhost:27017/")

mydb=myclient["mydb"]

positive\_vocab = ['good','Excellent','Punctual and Regular','Punctual but Irregular','Audible but Discontinuous','Clear and Audible','Organizing','Encouraging','High']

negative\_vocab = ['Poor','No','Not Punctual and Irregular','Neither Clear nor Audible','Spends Time with Irrelevant Talk']

neutral\_vocab = ['Avg','Medium','Fair','Regular but not Punctual']

positive\_features = [(word\_feats(pos), 'pos') for pos in positive\_vocab]

negative\_features = [(word\_feats(neg), 'neg') for neg in negative\_vocab]

neutral\_features = [(word\_feats(neu), 'neu') for neu in neutral\_vocab]

train\_set = negative\_features + positive\_features + neutral\_features

@app.route('/fedresult',methods=['GET','POST'])

def fedresult():

subj=request.json['sub']

name=request.json['name']

classifier = NaiveBayesClassifier.train(train\_set)

pos1=0

neg1=0

neu1=0

pos2=0

neg2=0

neu2=0

pos3=0

neg3=0

neu3=0

pos4=0

neg4=0

neu4=0

pos5=0

neg5=0

neu5=0

pos6=0

neg6=0

neu6=0

col=mydb[name]

num=col.find({} ,{subj})

result={}

for i in num:

for k in i:

if(k==subj):

print(k)

for n in i[k]:

for key, value in n.items():

if(key=="Explanation of Topic"):

result[key]={}

nm=result[key]

classResult = classifier.classify( word\_feats(value))

if classResult == 'neg':

neg1 = neg1 + 1

if classResult == 'pos':

pos1 = pos1 + 1

if classResult == 'neu':

neu1 = neu1 + 1

nm["pos"]=pos1

nm["neg"]=neg1

nm["neu"]=neu1

if(key=="Punctuality"):

result[key]={}

nm=result[key]

classResult = classifier.classify( word\_feats(value))

if classResult == 'neg':

neg2 = neg2 + 1

if classResult == 'pos':

pos2 = pos2 + 1

if classResult == 'neu':

neu2 = neu2 + 1

nm["pos"]=pos2

nm["neg"]=neg2

nm["neu"]=neu2

if(key=="Use of I.C.T"):

result[key]={}

nm=result[key]

classResult = classifier.classify( word\_feats(value))

if classResult == 'neg':

neg3 = neg3 + 1

if classResult == 'pos':

pos3 = pos3 + 1

if classResult == 'neu':

neu3 = neu3 + 1

nm["pos"]=pos3

nm["neg"]=neg3

nm["neu"]=neu3

if(key=="Clarity & Audibility"):

result[key]={}

nm=result[key]

classResult = classifier.classify( word\_feats(value))

if classResult == 'neg':

neg4 = neg4 + 1

if classResult == 'pos':

pos4 = pos4 + 1

if classResult == 'neu':

neu4 = neu4 + 1

nm["pos"]=pos4

nm["neg"]=neg4

nm["neu"]=neu4

if(key=="Use of Black Board"):

result[key]={}

nm=result[key]

classResult = classifier.classify( word\_feats(value))

if classResult == 'neg':

neg5 = neg5 + 1

if classResult == 'pos':

pos5 = pos5 + 1

if classResult == 'neu':

neu5 = neu5 + 1

nm["pos"]=pos5

nm["neg"]=neg5

nm["neu"]=neu5

if(key=="Seminars/ Projects"):

result[key]={}

nm=result[key]

classResult = classifier.classify( word\_feats(value))

if classResult == 'neg':

neg6 = neg6 + 1

if classResult == 'pos':

pos6 = pos6 + 1

if classResult == 'neu':

neu6 = neu6 + 1

nm["pos"]=pos6

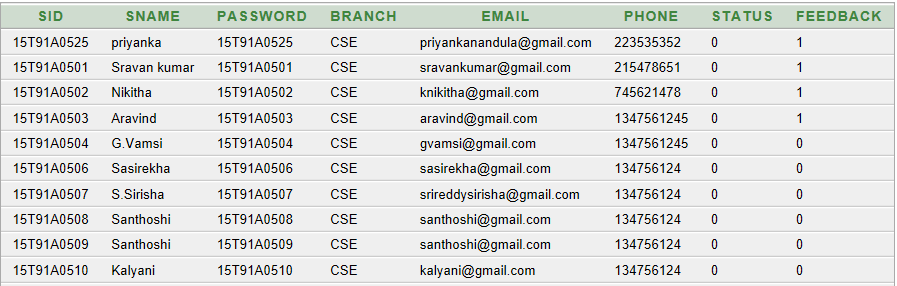
nm["neg"]=neg6

nm["neu"]=neu6

return render\_template('graph.html',data=result)

**5.3 DATABASE TABLES**

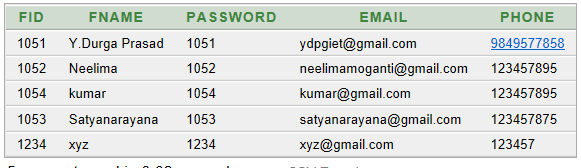
**5.3.1 STUDENT TABLE**

****

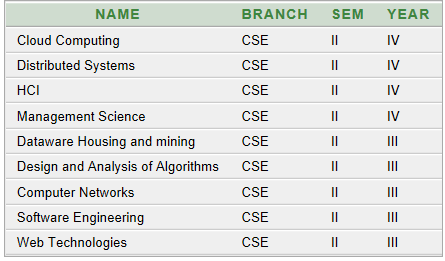
**5.3.2 ADMIN TABLE**

**ss3**

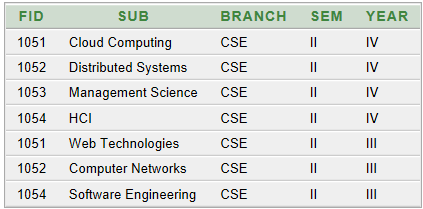
**5.3.3 FACULTY**

****

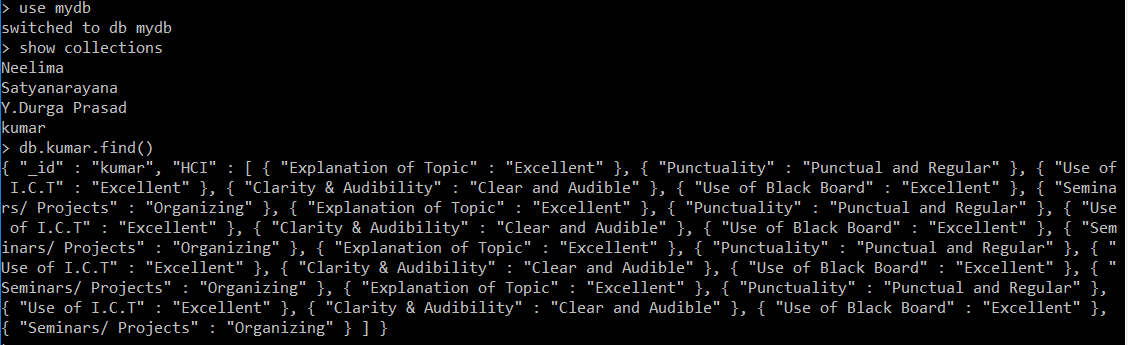
**5.3.4 SUBJECT**

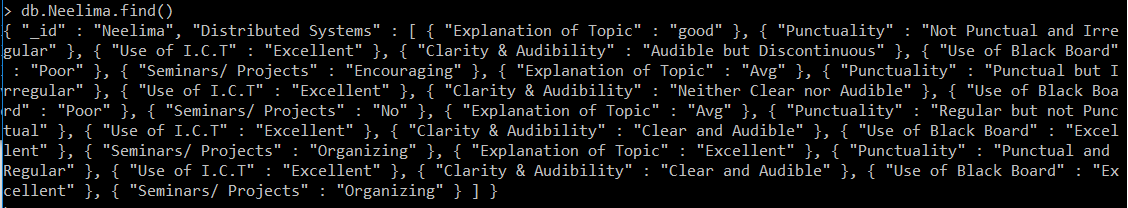
****

**5.3.5 SUBJECT DETAILS**

****

**5.3.6 FEEDBACK TABLE**

****

****

**6. SYSTEM TESTING**

**6.1 INTRODUCTION**

A process of executing a program with the explicit intention of finding errors, that is making the program fail. Testing is the process of detecting errors. Testing performs a very crucial role for quality assurance and for ensuring the reliability of software The results of testing are used later in during maintenance also.

Testing is required to make sure whether the functionality of the system is working or not. Testing is performed to ensure that software program complies to all the requirements gathered from the client or customer

**WHITE BOX TESTING:**

White box testing also known as box testing, glass box testing, transparent box testing, and structural testing is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality.

**BLACK BOX TESTING:**

Black box testing also known as Behavioral testing, is a software testing method in which the internal structure /design/implementation of the item being tested is not known to the tester. These tests can be functional or non functional though usually functional.

**UNIT TESTING:**

A software module can be created by building up of many small parts into a single module. This small part is called as a unit. A unit is a piece of code that will perform a specific task. At the end of this testing all units will be tested so that we can get the correct result. By using unit testing we can easily identify the errors.

**INTEGRATION TESTING:**

Combining all programs into a single application and testing its correct is called as Integration testing. Even all programs work correctly they may give a false result when they work together. Integration is very important to get the completed result.

**SYSTEM TESTING:**

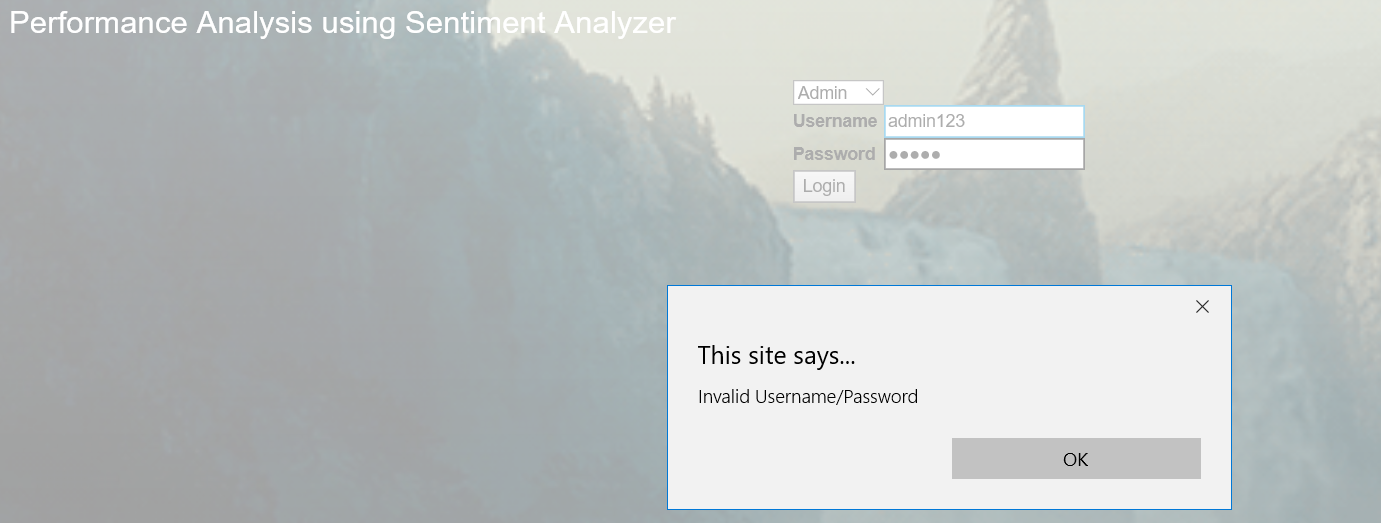
System testing means testing the whole system at once. By giving different inputs to the system we can check its correctness. For all inputs the system should produce correct result.

**REGRESSION TESTING:**

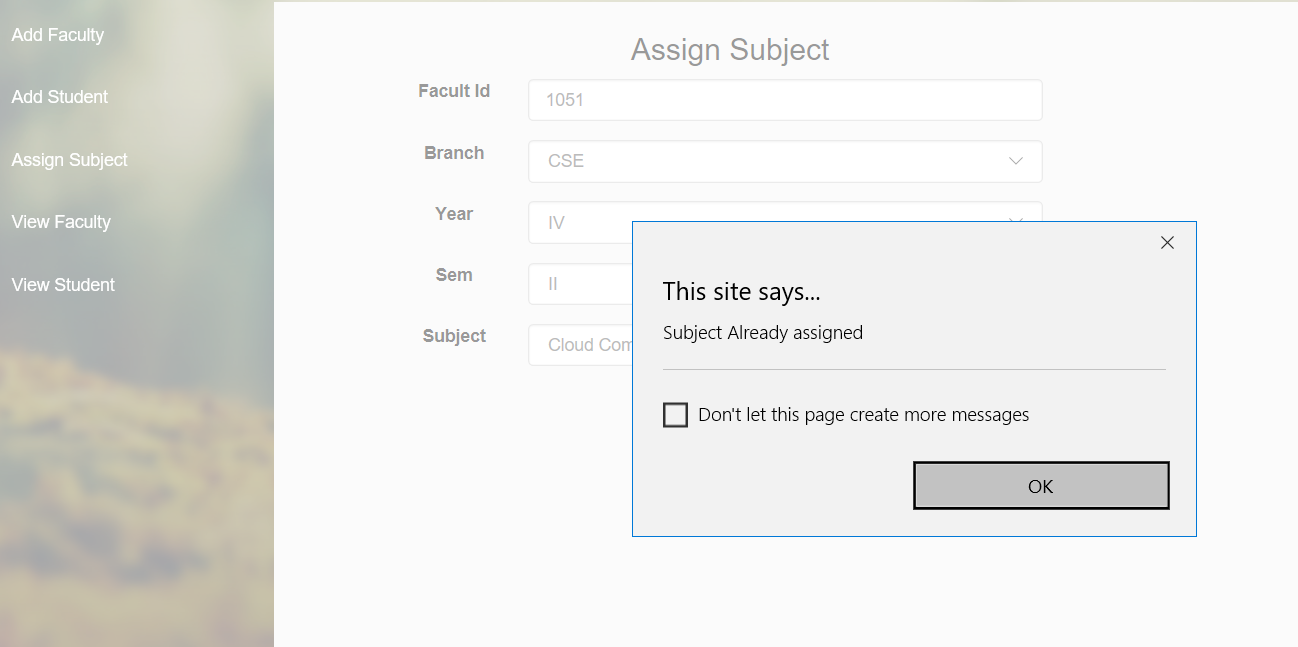
Regression Testing is nothing but full or partial selection of already executed test cases which are re-executed to ensure existing functionalities work fine.

This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

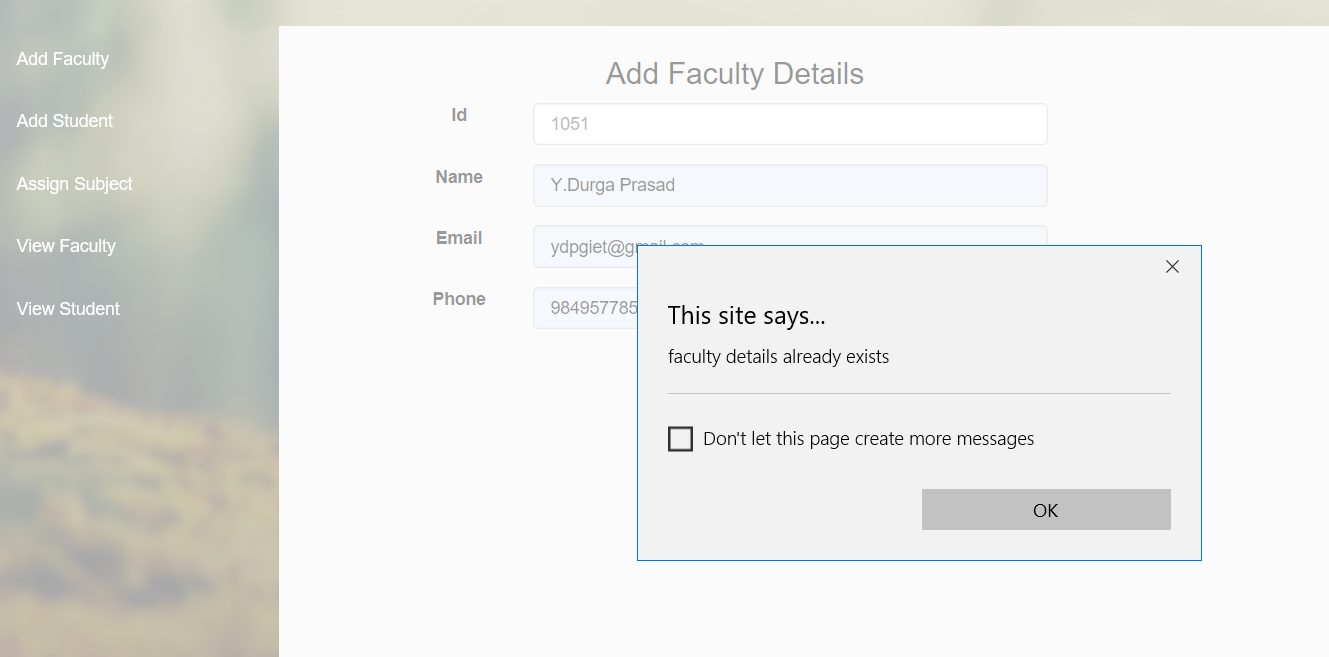
**6.2 TEST CASES**

****

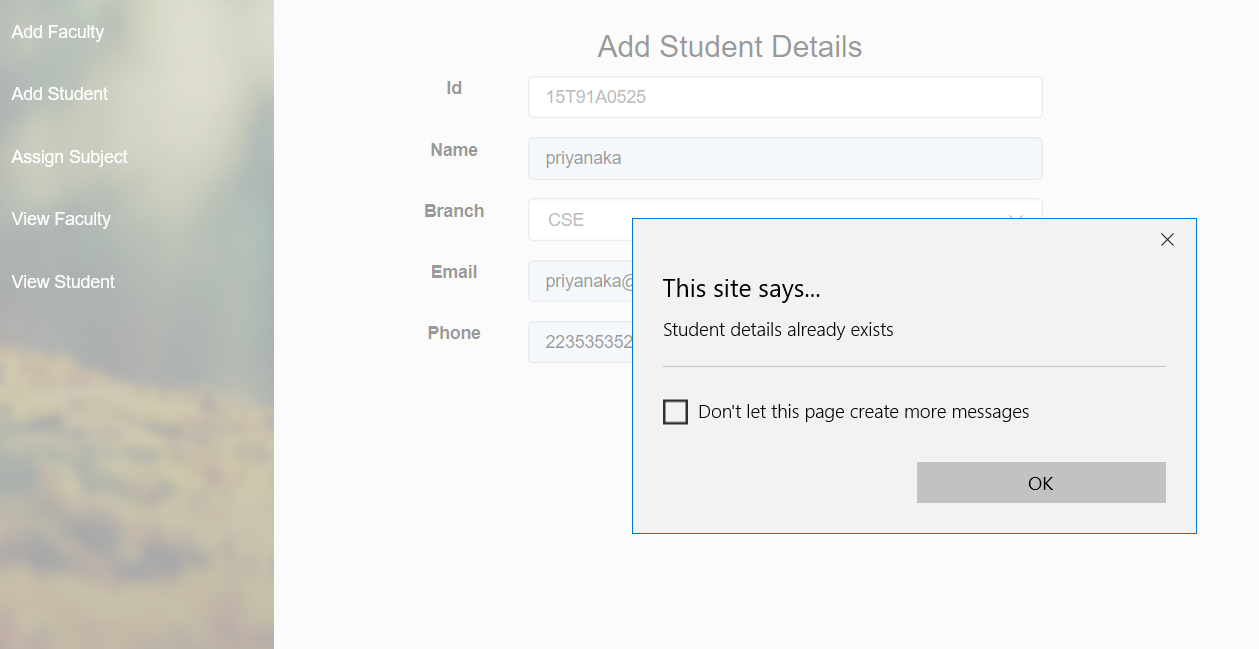
Screenshot-6.1 Testing Login Page



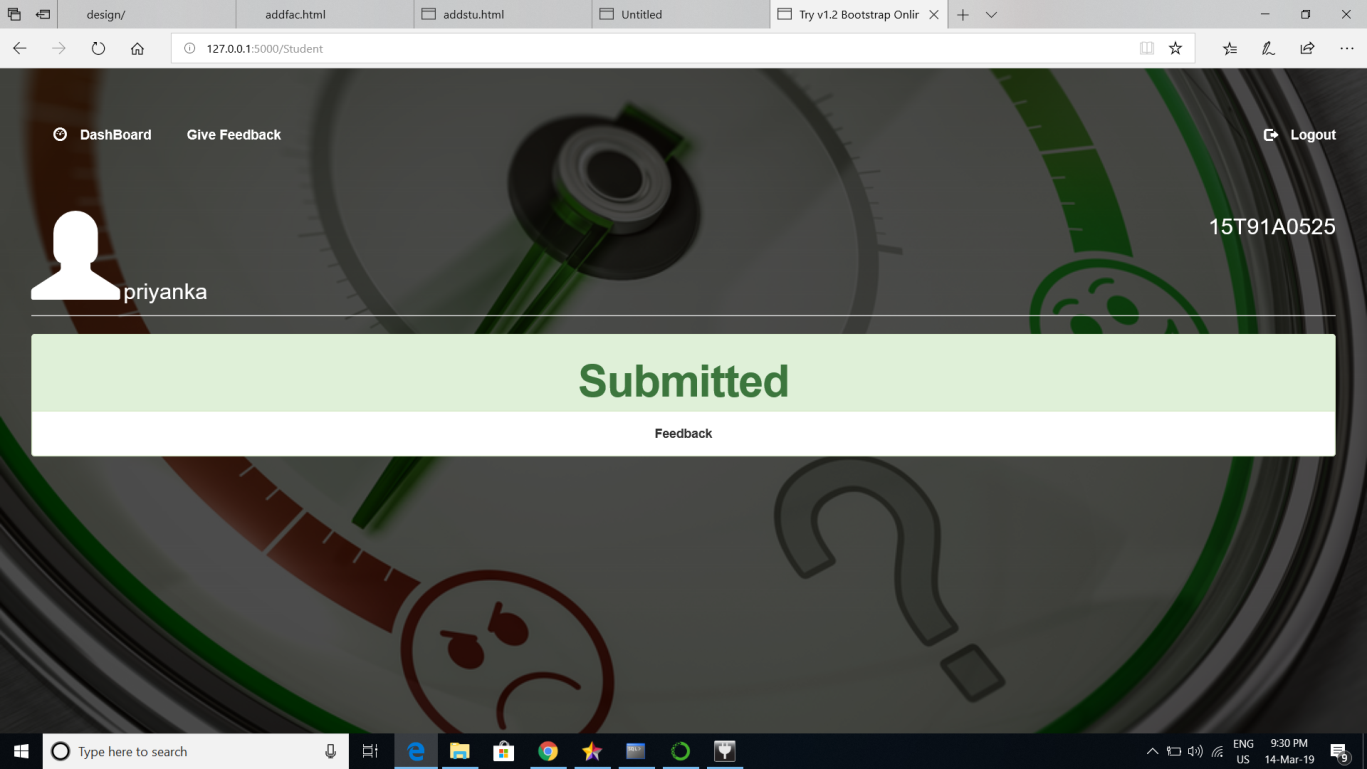
Screenshot-6.2 Testing Assign Subject Page



Screenshot-6.3 Testing Add Faculty Page



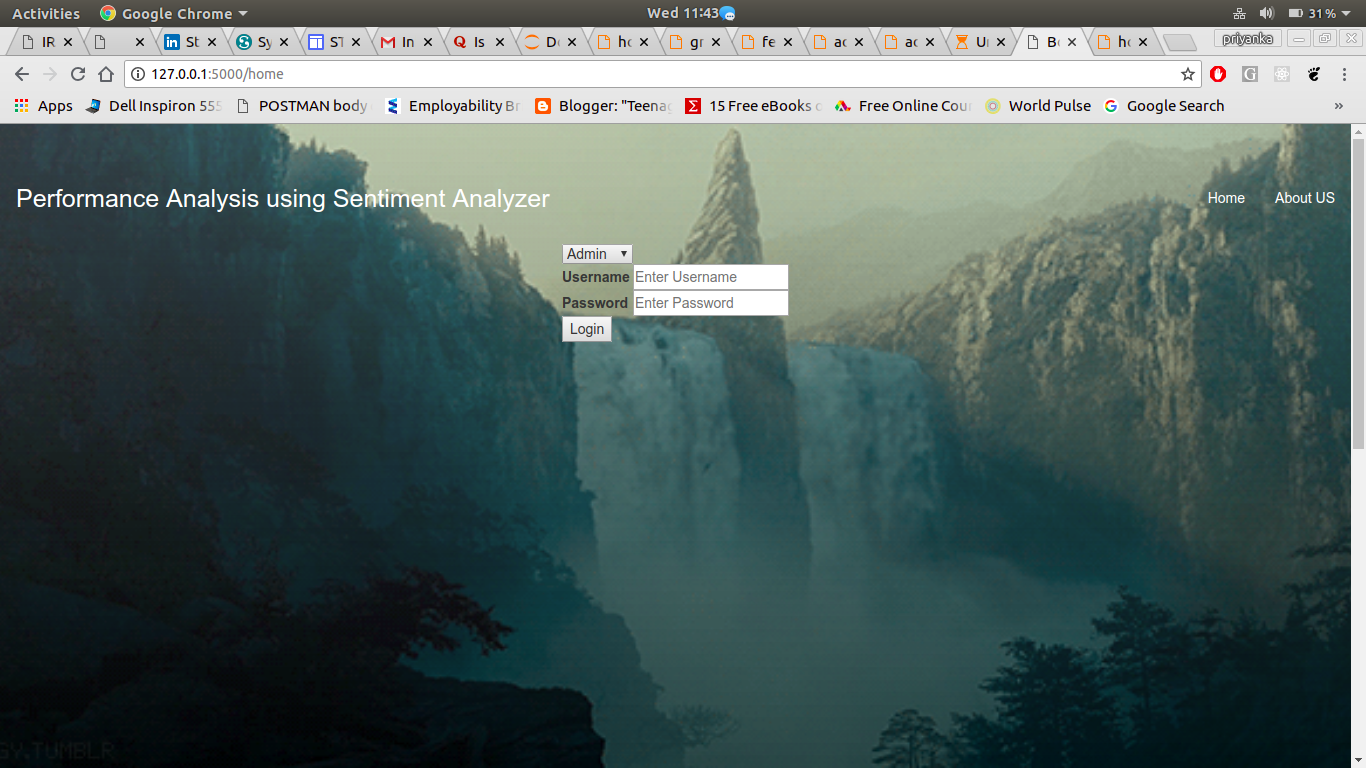
Screenshot-6.4 Testing Add Student Page



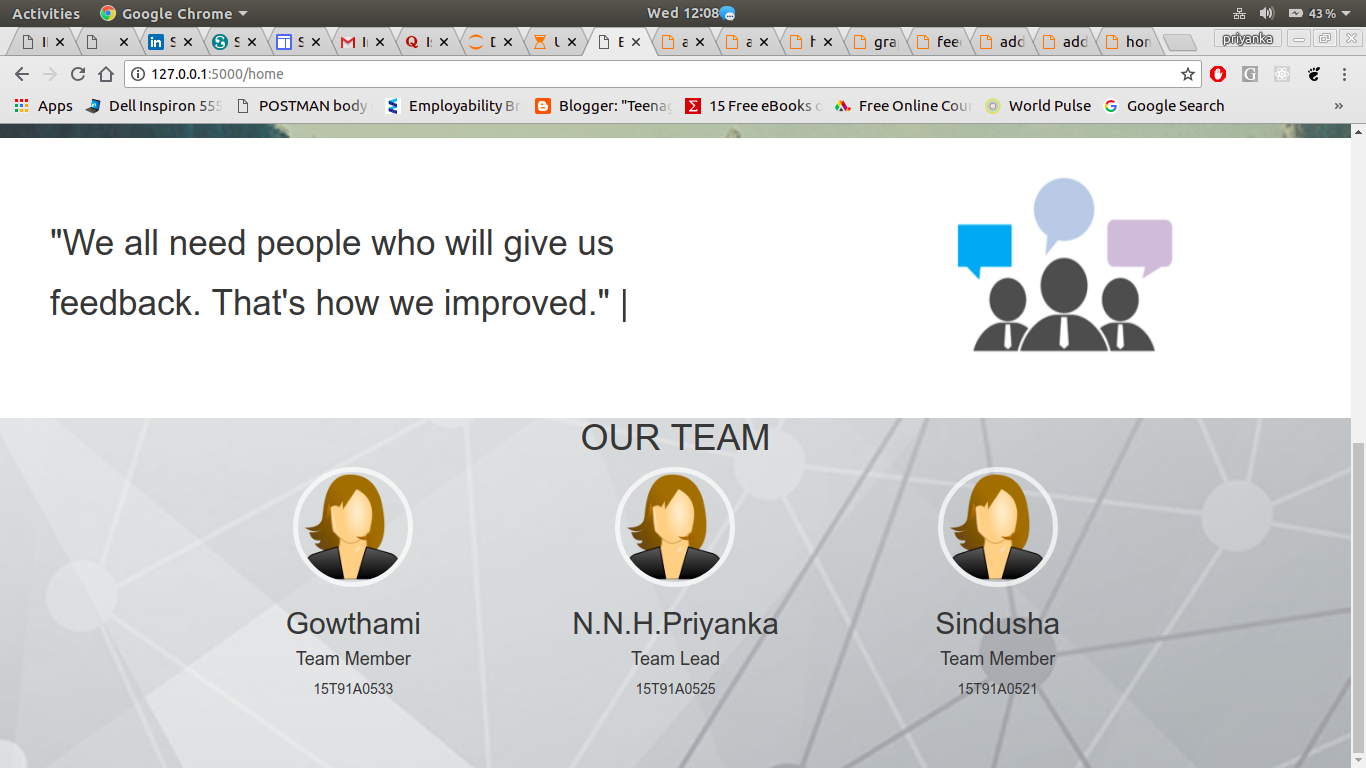
Screenshot-6.5 Testing Feedback Status Page

**7.SCREENSHOTS**

**HOME PAGE**

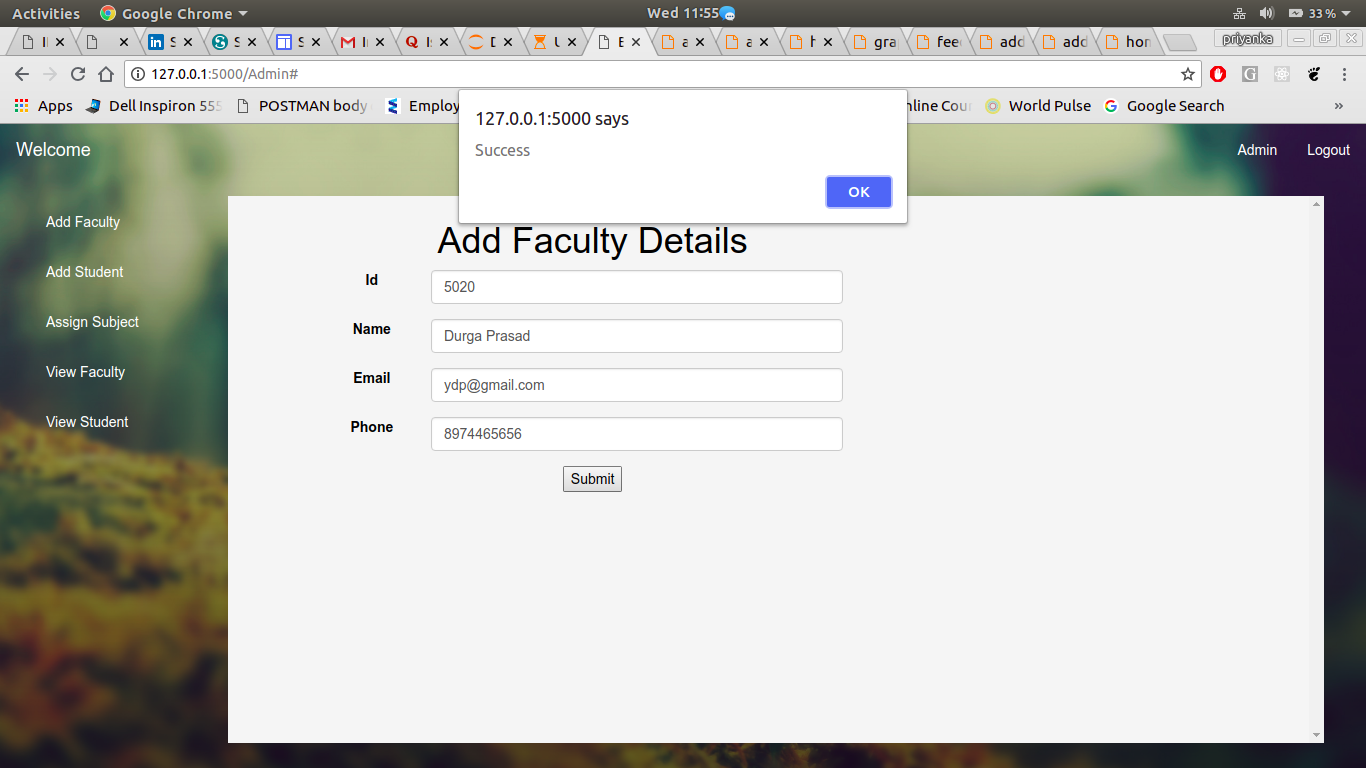


Screenshot 7.1 Home Page Screen 1

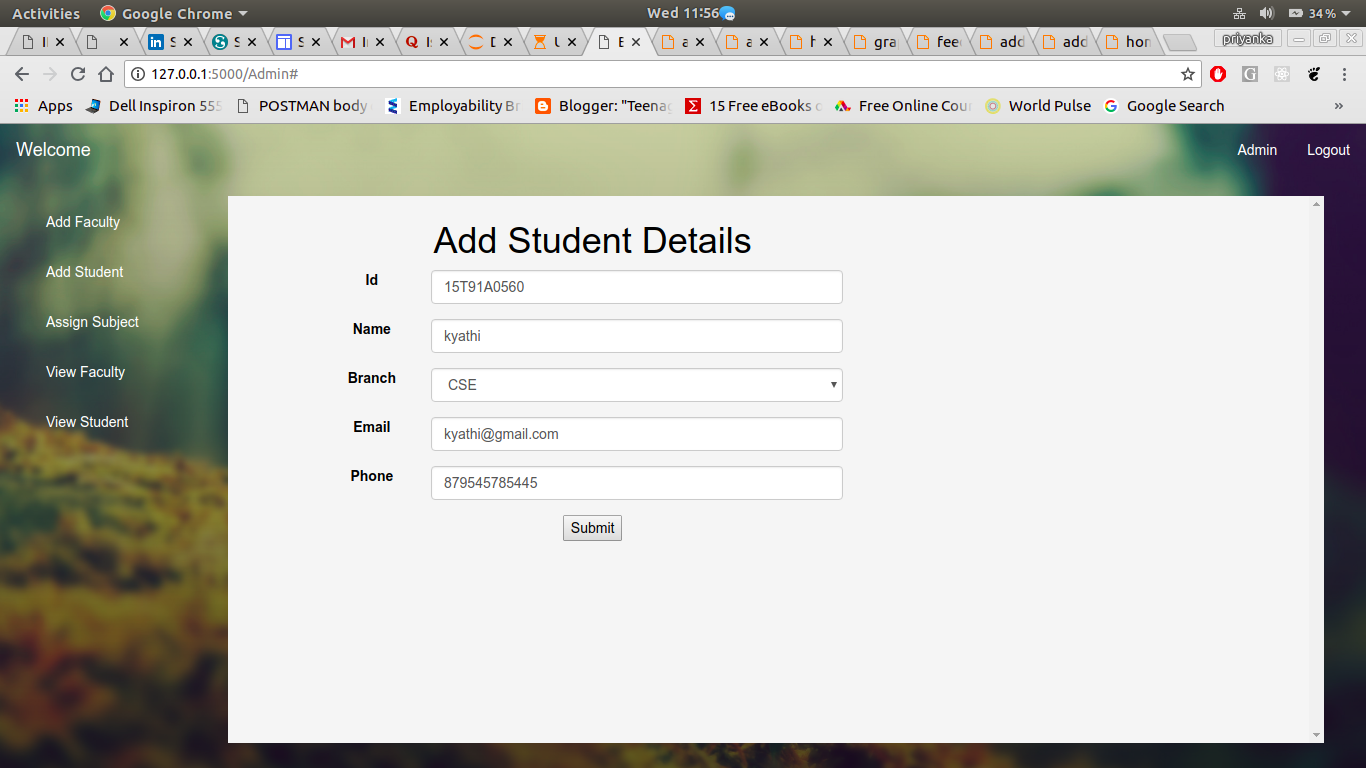


Screenshot 7.2 Home Page Screen 2

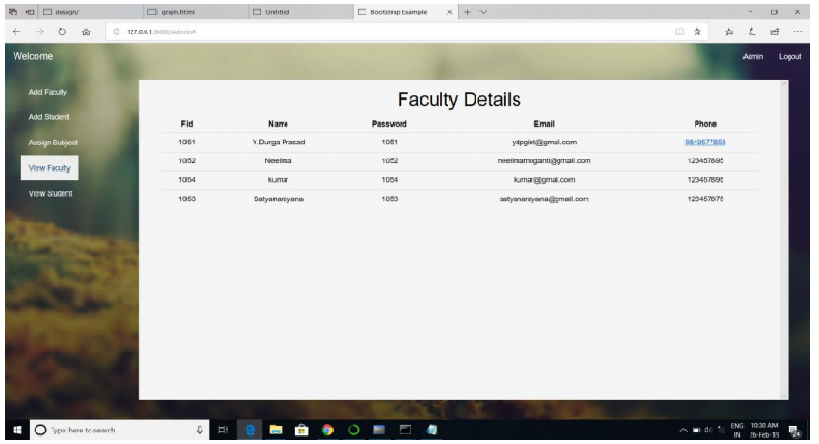
**ADMIN:**



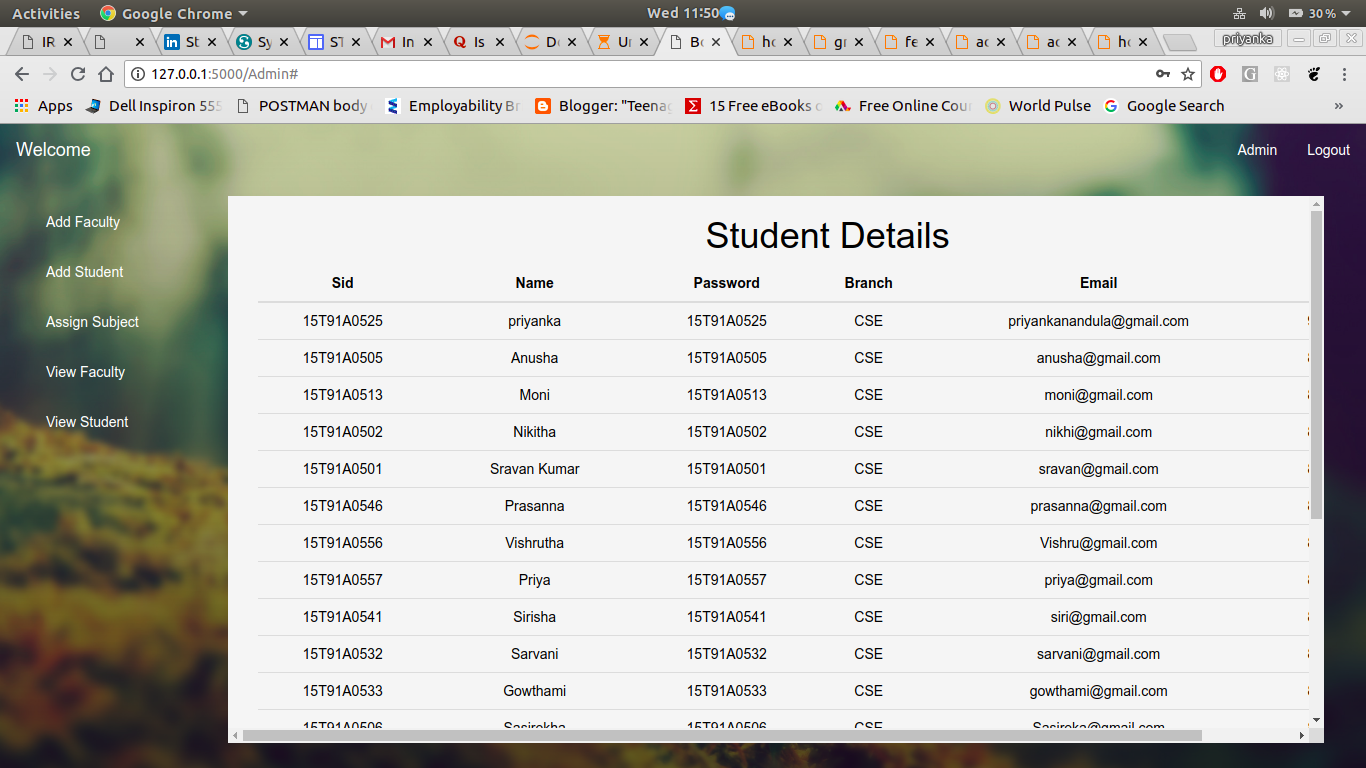
Screenshot 7.3 Add Faculty Page



Screenshot 7.4 Add Student Page

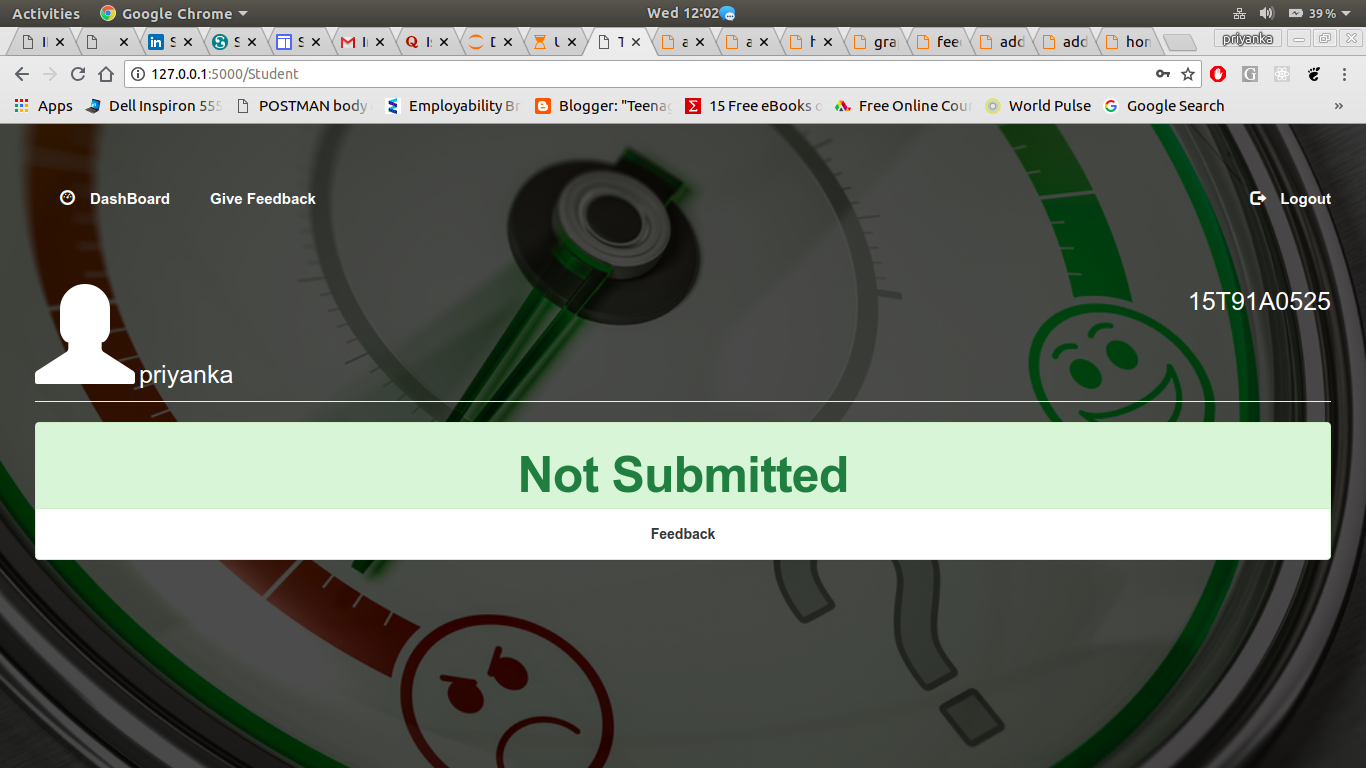
****

Screenshot 7.5 View Faculty Page

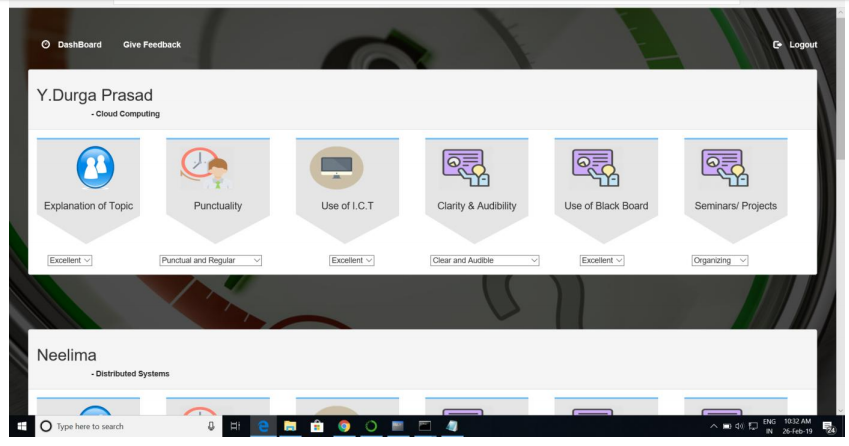


Screenshot 7.6 View Student Page

**Student**

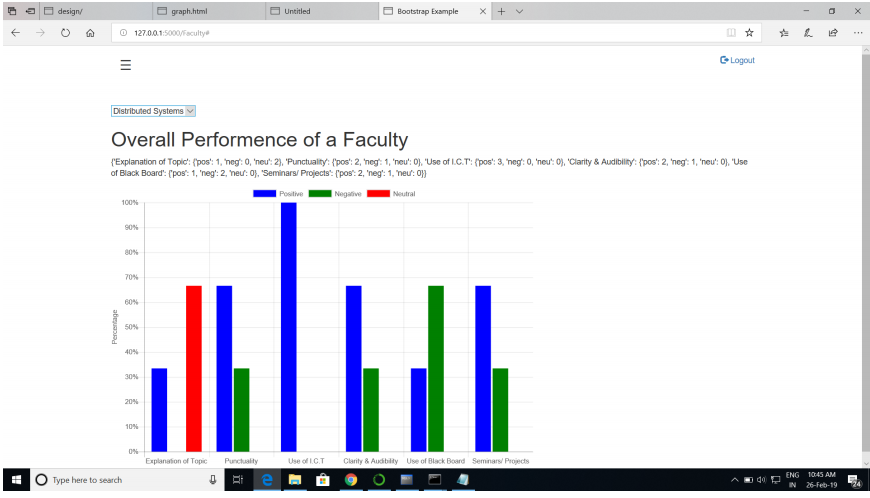


Screenshot 7.7 Dashboard Page

****

Screenshot 7.8 Give Feedback Page

**Faculty**

****

Screenshot 7.9 View Feedback Page

**8. CONCLUSION**

In this project a sentiment analyzer is implemented to analyze the student’s feedback so to increase the accuracy of the system. It is a way of determining how positive or negative the content of a text is, based on the relative numbers of words it contains that are classified as either positive or negative. Sentiment analysis tools can identify and analyse the text automatically and quickly.

**9.FUTURE ENHANCEMENT**

This project is beneficial for college. Teacher’s performance can be increased using the analyzed feedback. Reduces time – Giving feedback online saves time in comparison to manual process. Security – The identity of the student is kept anonymous. False feedback can be prevented from submission. Descriptive type questions can be analyzed by using the sentiment analyzer which was not possible in previous online feedback analysis system. Due to lack of time design part is not so attractive to look. Further enhancement can be made in the designing part. Some more forms can also be added so as to better retrieve the feedback details. Various other options can also be added for the better usability of the project. More positive and negative words can be added in the database for increasing the accuracy of the sentiment analyzer. This feedback system and sentiment analyzer can also be used in other organizations, offices for taking feedbacks.

**10.BIBILOGRAPHY**

* WEB TECHNOLOGIES the Complete Reference by ACHYUTH GODBOLE
* HTML & CSS: The Complete Reference, Fifth Edition by THOMAS POWELL
* [Machine Learning For Dummies” by John Paul Mueller and Luca Massaron](https://www.amazon.com/Machine-Learning-Dummies-John-Mueller-ebook/dp/B01FL4BGOI/)
* [Machine Learning in Action” by Peter Harrington](https://www.amazon.com/gp/product/1617290181/)
* Natural Language Processing with Python by SteveBird, EwanKlein and Edward Loper.
* Taming Text by [Grant Ingersoll](https://www.linkedin.com/in/grantingersoll/), [Thomas Morton](https://www.linkedin.com/in/tom-morton-5248552/) and [Drew Farris](https://www.linkedin.com/in/drewfarris/).

**11. WEBILOGRAPHY**

https://www.tutorialspoint.com/flask

<https://www.w3schools.com/jquery/>

<https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>

<https://pythonspot.com/python-sentiment-analysis/>

<https://www.w3schools.com/python/python_mongodb_getstarted.asp>