Summer Project report on

# College Review Sentiment Analysis

**By**

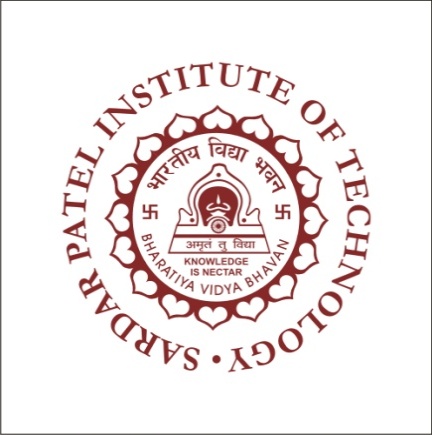
# Ashish A. Mishra

**Ghanshyam R. Gupta**

Under the guidance of

**Internal Supervisor**

**Prof. Mayura Nagar**

****

Department of Master In Computer Applications

Sardar Patel Institute of Technology

University of Mumbai

2016-17

**CERTIFICATE OF APPROVAL**

This is to certify that the following students

**Ashish A. Mishra**

**Ghanshyam R. Gupta**

Have satisfactorily carried out work on the project entitled

# “College Review Sentiment Analysis”

Towards the fulfillment of summer project, as laid down by University of Mumbai during year 2016-17.

Project Guide

**PROJECT APPROVAL CERTIFICATE**

This is to certify that the following students

**Ashish A. Mishra**

**Ghanshyam R. Gupta**

Have successfully completed the Project report on **“College Review Sentiment Analysis*”***, which is found to be satisfactory and is approved

At

SARDAR PATEL INSTITUTE OF TECHNOLOGY,

ANDHERI (W), MUMBAI.

INTERNAL EXAMINER EXTERNAL EXAMINER

Head of Department Principal

# Table of contents

|  |  |  |
| --- | --- | --- |
| **Serial no.** | **Topic** | **Page no.** |
|  | Abstract……………………………………………….................. | i |
|  | Objectives……………………………………………………….. | ii |
|  | List of figures……………………………………….................... | iii |
|  | List of tables………………………………………….................. | iv |
| **1** | **Introduction……………………………………….....................** | **1** |
| 1.1 | Problem Definition………………………………….................... | 1 |
| 1.2 | Objective and Scope………………………………….................. | 1 |
| 1.3 | Existing System……………………………………..................... | 1 |
| 1.4 | Proposed System……………………………………................... | 1 |
| 1.5 | System Requirements………………………...………................. | 2 |
| **2** | **Literature Survey………………………………………………** | **3** |
| 3 | **SRS and Design…………………………………………………** | 5 |
| 3.1 | Introduction………………………………………….................. | 5 |
| 3.1.1 | Purpose………………………………………………................. | 5 |
| 3.1.2 | System Overview……………………………………................. | 5 |
| 3.2 | Overall Description………………................………………….. | 5 |
| 3.2.1 | Product functions.................……………………………………. | 5 |
| 3.2.2 | User characteristics.................………………………………….. | 5 |
| 3.3 | Specific Requirements.................………………………………. | 6 |
| 3.3.1 | External Interface Requirements................…………………….. | 6 |
| 3.3.3 | Logical DB Requirements................…………………………… | 6 |
| **4** | **Project Analysis and Design…………………………………..** | **7** |
| 4.1 | Methodologies Accepted.................……………………………. | 7 |
| 4.1.1 | Detailed life cycle of Project................………………………… | 7 |
| 4.2 | UML Diagrams..................……………………………………… | 10 |
| 4.2.1 | Use case Diagram..................…………………………………… | 10 |
| 4.4 | Database Design.................……………………………………... | 16 |
| **5** | **Project Implementation and Testing.................………………** | **18** |
| 5.1 | Work Breakdown Structure.................………………………….. | 18 |
| 5.2 | Gantt Chart.................…………………………………………... | 20 |
| 5.3 | Pert Chart.................…………………………………………….. | 21 |
| 5.4 | Timeline chart.................………………………………………... | 22 |
| 5.5 | Code with reference to design.................……………………….. | 23 |
| 5.6 | Snapshot of UI.................……………………………………….. | 30 |
| 5.7 | Test Cases and Report.................……………………………….. | 32 |
| 5.7.1 | Types of Testing….................…………………………………... | 32 |
| 5.7.1.1 | White Box Testing.................…………………………………… | 32 |
| 5.7.1.2 | Black Box Testing.................…………………………………… | 32 |
| 5.7.2 | Test Report.................…………………………………………... | 33 |
| **6** | **System Maintenance & Evaluation…………….……………...** | **34** |
| 6.1 | Maintenance..................…………………………………………. | 34 |
| 6.2 | Evaluation..................……………………………………………. | 34 |
| **7** | **Future Enhancements……………………………..……………** | **35** |
| **8** | **Limitations………………………..……………………………..** | **36** |
| **9** | **Conclusion…………………………………………………….....** | **37** |
| **10** | **Bibliography…………………………………………………..…** | **38** |

Abstract

This project deals with developing a Web application for College Review System using sentiment Analysis. It Provides user to get the clear knowledge about the respective college, courses etc. One can search for different colleges. It is important to have reviews for any college in terms of placement, faculties, infrastructure etc. Our system majorly focuses on review parts. The system is implemented using Data mining which will predict positive and negative responses given by user to the college now with the Help of Weka Tool it will give details to us whether college is good or bad. A graph chart will be shown to the user with the results.

Our application uses front end as JSP, with backend in Servlets and MySQL database.

**Objectives**

The objective of the project is to develop a web application which can be useful to the users.

To get details about college

To make them access prediction analysis while adding reviews

To show the results based on review given by user

1. **Introduction**

In Sentiment Analysis for College Review System using Data Mining, the user will be able to add reviews for college. The system will be able to give prediction to the user on the basis of added reviews. A user can also be given an option to predict what they have given in review. The website has a feature to predict responses given by user to give appropriate results.

**1.1 Problem Definition**

The website uses the JAVA and HTML languages for its development

The term project is to design and implement a Web application for Sentiment Analysis for College Review System using Data Mining.

The College Review System will provide option, users to get registered first after which user will be given username and Password. After logging in a welcome message containing the name, must be displayed for all users.

Users: After logging in, the user will be able to view the short brief of about web application. They will also be able to search for Colleges and the number of The shopping cart page will contain a summary of the order. The Users will have facility where the most frequently added reviews will be shown to them.

**1.2 Objective & Scope**

**Objective**

To get details about college

To make them access prediction analysis while adding reviews

To show the results based on review given by user

**Scope**

Primarily, the scope pertains to Read or Write new review features. It focuses on the prediction methods, and applications.

This project provides various facilities like searching for a college, facility to view the most popular college based on its review etc.

**1.3 Existing System**

In the existing system for getting good feedback user needs to visit a college and sometimes they look for a help on internet, there are no recommendations to the users according to their search record. Also, there is no such feature as the most popular colleges.

**1.4 Proposed System**

The proposed system gives recommendations to the user based on their review records. It also has a feature that shows which college is the most popular. So this system will help user to know about college efficiently.

The system has a feature where user can add reviews of his/her own choice now system will predict those reviews along with existing reviews and gives a result.

**1.5 System Requirements**

Hardware:

Processor Intel(R) Pentium(R) D CPU 2.66 GHz

RAM 512 MB

Operating System Windows XP

Software:

Front End Editor

XAMPP

Browser

1. **Literature Survey**

Many research has shown students find it difficult to get the importance of college.

The most cited reason for using the web for College review was convenience (65%)

A good review system should present the following factors to the users for better usability:

· Knowing when a review details was saved or not saved in system.

· Easy scanning and selection of college.

· Effective categorical organization of products

. · Simple navigation from home page to information.

· Obvious shopping links or buttons.

**3. Software Requirement Specification[SRS] & Design**

**3.1 Introduction**

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete Book buying system by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the Book buying portal are provided in this document.

**3.1.1 Purpose**

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to users. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

**3.2 Overall description**

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of it. The following SRS contains the detail product perspective from different stakeholders. It provides how one can make best out of it.

**3.2.1 Product Functions**

There will be registration for every user. They will be given a login access. The Web application will be able to give options to the users to search a college, get the details of the existing college and can add their own reviews on it.

**3.2.2 User Characteristics**

The User will be able to use all the functionality of the web application like searching for colleges, Adding reviews, etc.

**3.3** **Specific requirements**

**3.3.1 Non-Functional Requirements**

**3.3.1.1 Performance requirements:**

The Software should be web based and it should perform all the operations regarding to search and access of the database to increase the efficiency of the system.

**3.3.1.2 Safety requirements:**

### The Authentication should be safe enough to use. No other third party person should be able to add data or change the information of the system.

The requirement for making the system safe enough is to make the database password protected.

**3.3.2 Security Requirements:**

### **3.3.2.1 Data Transfer**

The system shall use secure sockets that include any confidential User information.

### **3.3.2.2 Data Storage**

The user’s terminal shall never display a user’s password. It shall always be echoed with special characters representing typed characters.

The system’s back-end servers shall never display a user’s information.

**3.3.3 Quality requirement**

**3.3.3.1 Software Quality Attributes**

Quality attribute requirements are part of an application’s nonfunctional requirements, which capture the many facets of how the functional requirements of an application are achieved. All but the most trivial application will have nonfunctional requirements that can be expressed in terms of quality attribute requirements. To be meaningful, quality attribute requirements must be specific about how an application should achieve a given need. A common problem I regularly encounter in architectural documents is a general statement such as “The application must be scalable. In **Sentiment Analysis of College Review using Data mining,** The customer should be able to search colleges properly. The system should be able to ask for sentiment analysis suggestions to the user on the basis of what they give review. If it is so then the software is of good quality. Depending on the attributes of the software ,our projects quality is measured.

* 1. **Other requirements**

### **Graphical User Interface**

The system shall provide a uniform look and feel between all the web pages.

The system shall provide a digital image for each product in the product catalog.

The system shall provide use of icons and toolbars.

1. **Project Analysis and Design**

**4.1 Methodologies Adapted**

To give recommendations to the user we fired query specific to it. We used an attribute in the table whose value will increase whenever any user views a particular book. On the basis of the views, we suggest users the most Frequently viewed products. And to show an user the most popular product, we wrote a query which retrieves the maximum of buy attribute from the table and displays a book which has the same Book ID as that of maximum bought book’s.

**4.1.1 Detailed life Cycle of the Project**

* + 1. Detailed life Cycle of the Project
* 1st release:

The product will have login page and registration screen.

* 2nd release:

The system will have home page created along with search option to look up college information.

* 3rd release:

Data mining and machine learning algorithms will be used with the help of Weka for predicting.

* 4th release:

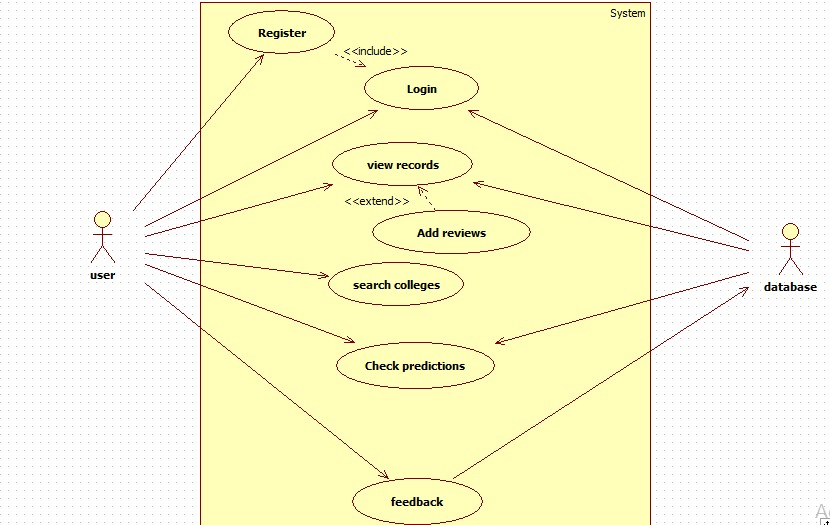
Database will be created for user authentication and storing college details.

* 5th release:

User feedback mechanism is added and graphs are included to summarize the sentiment of the college.

**4.2 UML Diagram**

**4.2.1 Use Case Diagram with Report**



**4.4.3 Database schema design**

**Database Name: books**

**College Details Table**

|  |  |  |
| --- | --- | --- |
| Sr No | Name | Type |
| 1 | Id | Int |
| 2 | Name | Varchar |
| 3 | Address | Varchar |
| 4 | Contact | Varchar |
| 5 | Website | Varchar |
| 6 | Negative | Int |
| 7 | Positive | Int |
| 8 | Positivity | Int |

**College Reviews table**

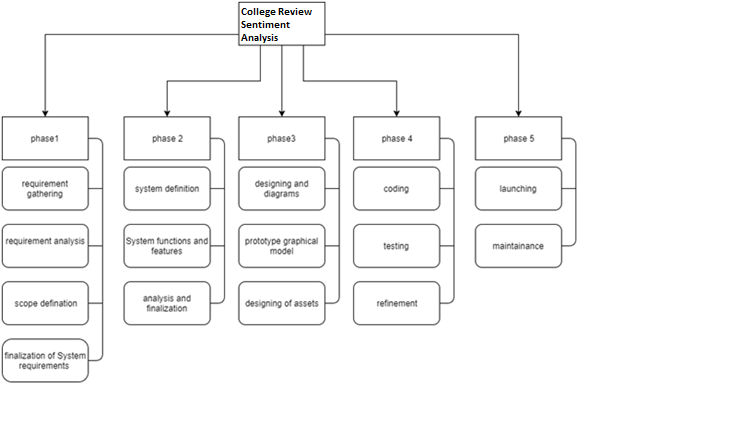
|  |  |  |
| --- | --- | --- |
| Sr No | Name | Type |
| 1 | User | Varchar |
| 2 | Name | Varchar |
| 10 | Review | Varchar |

**Feedback details table**

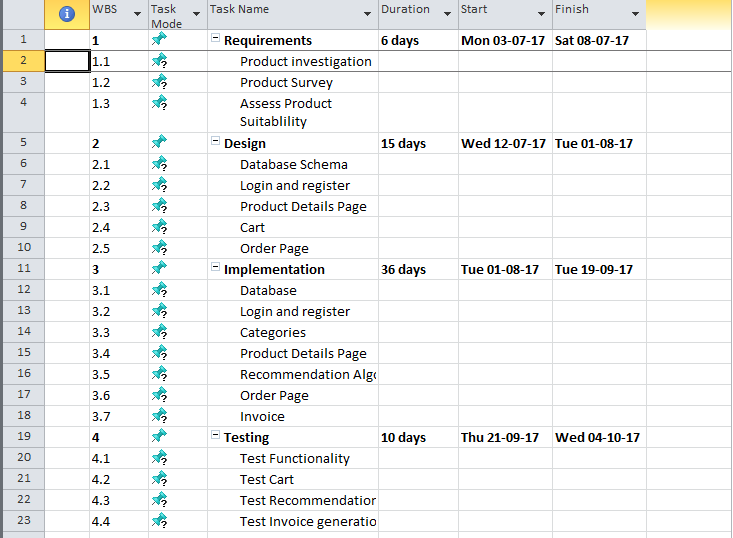
|  |  |  |
| --- | --- | --- |
| Sr No | Name | Type |
| 1 | Username | Varchar |
| 2 | EmailId | Int |
| 3 | Feedback | Varchar |

**5. Project Implementation & Testing**

**5.1 Work Break Down Structure**

****

**5.2 Gantt Chart**



**5.6 Code with reference to design**

**Code predicting sentiment:**

try {

//out.print("Analyzing review...");

// Instances class is used to select a dataset. The constructor takes the file path as parameter.

BufferedReader br = new BufferedReader(new FileReader(filePath));

Instances trainingData = new Instances(br);

trainingData.setClassIndex(trainingData.numAttributes() - 1);

br.close();

ArrayList<String> classVal = new ArrayList<>(2);

classVal.add("neg");

classVal.add("pos");

Attribute attr1 = new Attribute(trainingData.attribute(0).name(), (ArrayList<String>)null);

Attribute attr2 = new Attribute(trainingData.attribute(1).name(), classVal);

ArrayList<Attribute> attributes = new ArrayList<>(trainingData.numAttributes());

attributes.add(attr1);

attributes.add(attr2);

Instances inst = new Instances("Test", attributes, 1);

inst.setClassIndex(inst.numAttributes() - 1);

DenseInstance dInst = new DenseInstance(inst.numAttributes());

dInst.setValue(attr1, comment);

inst.add(dInst);

// System.out.println(trainingData);

// setClassIndex() is used to refer the column no. which has the answer.

StringToWordVector stwv = new StringToWordVector();

stwv.setInputFormat(trainingData);

stwv.setIDFTransform(true);

stwv.setTFTransform(true);

stwv.setLowerCaseTokens(true);

stwv.setOutputWordCounts(true);

// Creating a classifier, in my case, im using Support Vector Machines.

FilteredClassifier fc = new FilteredClassifier();

fc.setFilter(stwv);

fc.setClassifier(new SMO());

fc.buildClassifier(trainingData);

// Now im building that classifier from my training dataset.

// My 'testdata' has user-entered values(see above)).

// Im passing it to the classifier to get prediction.

double prediction = fc.classifyInstance(inst.instance(0));

out.print(inst.classAttribute().value((int)prediction));

// Here Im just displaying my predictions using a Dialog box.

//JOptionPane.showMessageDialog(this, inst.classAttribute().value((int)prediction), "Result", JOptionPane.INFORMATION\_MESSAGE);

} catch(Exception ex) {

out.print("Error Occurred: " + ex.toString());

//JOptionPane.showMessageDialog(this, ex.getMessage(), ex.getClass().toString(), JOptionPane.ERROR\_MESSAGE);

}

Javascript:

function getCollegeDetails(name) {

var xmlhttp = new XMLHttpRequest();

$('.tp').removeClass('hide');

$('#label').removeClass('hide');

$(".container").removeClass("hide");

xmlhttp.onreadystatechange = function ()

{

if (this.readyState === 4 && this.status === 200) {

$(".article").addClass("hide");

$("#contactPage").addClass("hide");

//document.getElementsByClassName("jumbotron")[0].innerHTML = "CollegeDB | A place to share your views";

data = this.responseXML;

var neg = data.getElementsByTagName("neg")[0].childNodes[0].nodeValue;

var pos = data.getElementsByTagName("pos")[0].childNodes[0].nodeValue;

cName = data.getElementsByTagName("name")[0].childNodes[0].nodeValue;

cSAddr = data.getElementsByTagName("saddr")[0].childNodes[0].nodeValue;

cAddr = data.getElementsByTagName("addr")[0].childNodes[0].nodeValue;

cWeb = data.getElementsByTagName("web")[0].childNodes[0].nodeValue;

cContact = data.getElementsByTagName("contact")[0].childNodes[0].nodeValue;

document.getElementById("cname").innerHTML = cName;

document.getElementById("saddr").innerHTML = cSAddr;

header = "Location";

title = cName;

text = cAddr;

document.getElementById("caddr").innerHTML = updateCardVal();

courses = data.getElementsByTagName("course");

header = "Courses Offered";

title = "";

course = "<ul>";

// course = "<b>Courses Offered:</b><br /><br />" + "<ul>";

for (i = 0; i < courses.length; i++) {

course += "<li>" + courses[i].childNodes[0].nodeValue + "</li>";

}

course += "</ul>";

text = course;

document.getElementById("cbranch").innerHTML = updateCardVal();

reviews = data.getElementsByTagName("review");

header = "Anonymous User";

title = "";

var top = "";

for (i = 0; i < reviews.length; i++) {

text = "<p style='padding:10px' id='" + (++commentNo) + "'>" + reviews[i].childNodes[0].nodeValue + "</p>";

top += updateCardVal();

}

document.getElementById("creviews").innerHTML = top;

google.charts.load('current', {'packages': ['corechart']});

google.charts.setOnLoadCallback(drawChart);

// Draw the chart and set the chart values

function drawChart() {

var data = google.visualization.arrayToDataTable([

['Sentiment', 'No of Review'],

['Negative Reviews', 2.5 \* neg],

['Positive Reviews', 2.5 \* pos]

]);

// Optional; add a title and set the width and height of the chart

var options = {

title: 'Review Summary',

width: 500, 'height': 350,

slices: {

0: {color: 'Red'},

1: {color: 'Green'}

}

//sliceVisibilityThreshold: 0.5

};

// Display the chart inside the <div> element with id="piechart"

var chart = new google.visualization.PieChart(document.getElementById('piechart'));

chart.draw(data, options);

}

} else {

//document.getElementById("cdetail").innerHTML = "LOADING...";

}

};

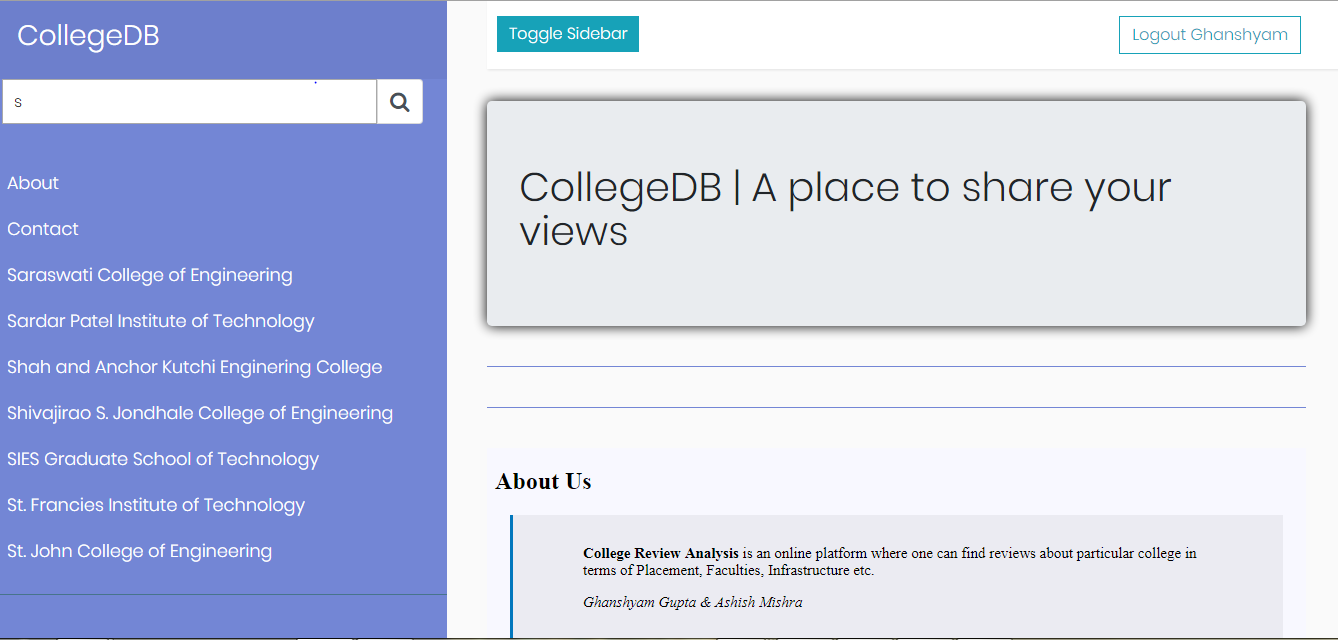
xmlhttp.open("POST", "http://localhost:8080/CollegeReview/CollegeDetails?collegeName=" + name, true);

xmlhttp.send();

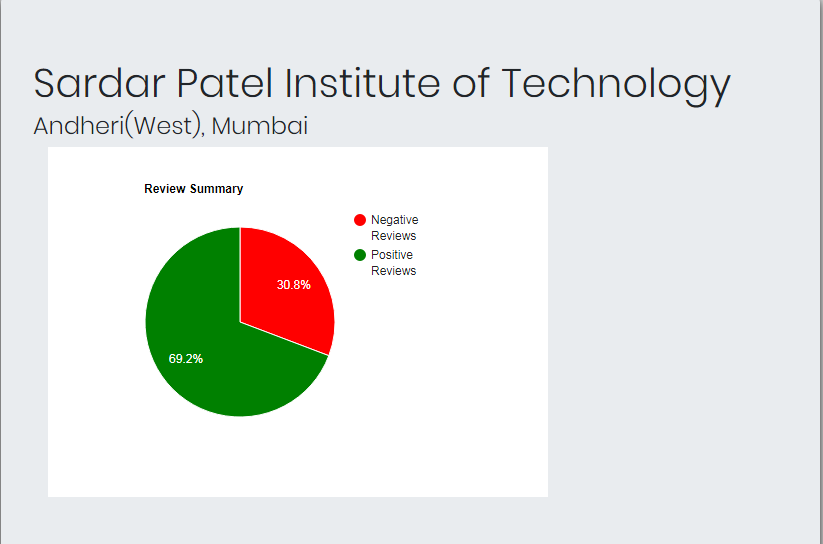
}

**5.7 Snapshot of UI & Reports**

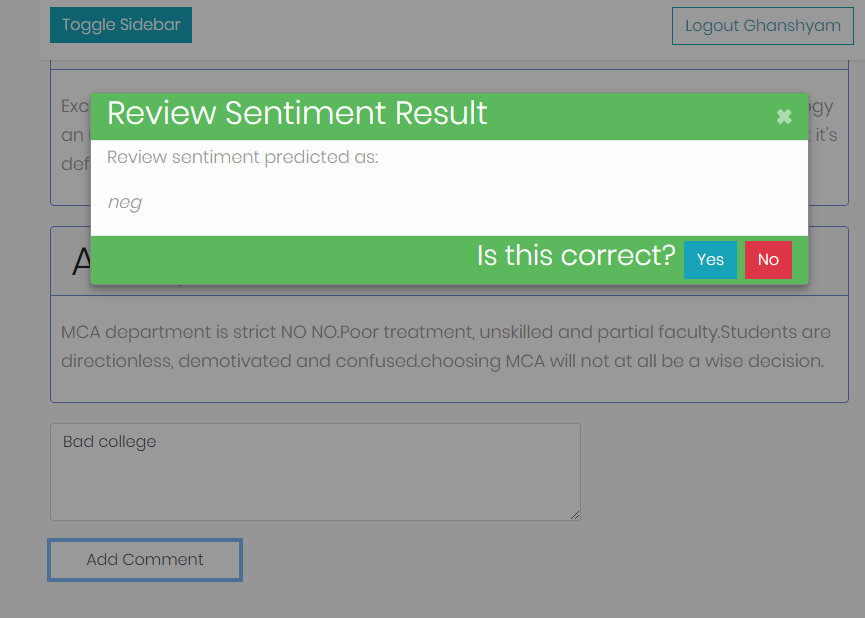
**Home page:**



**Review Graph:**



Sentiment Prediction:



**5.8 Test cases & Report**

**5.8.1 Types of Testing**

**5.8.1.1 White Box Testing**

White box testing is tough! Often our backgrounds prevent us from understanding the value of white box testing, as most testers don’t have programming experience, training and skills. Persisting and learning the techniques of white box testing can pay off in better test coverage and effective test design due to better understanding of the System Under Test (SUT) and the techniques that may be used to test it efficiently.

What is it?

White Box testing is using our skills to look into the structure of the System Under Test (SUT), either at the detailed design level or at the code level, to analyse and develop understanding, then to use this understanding to help us design and develop test cases. These test cases would focus on any potential issues we see in the detail of the SUT layout. These may be more and different risk areas or issues that can be identified than simply using the Black Box test techniques. We are basically using the structure of the system, in a similar way that we use the requirements, to tell us what tests to write and run. We can use the design and the code to achieve greater coverage of the system, and more specific tests focussed on high risk areas of the SUT.

Why do we do it? Testing is all about telling people about the quality of the system under test. That is our role. We focus our testing on various test levels or test for specific quality risks and tell people about the defects and behaviour of the system to ensure that the project knows what the system can and cannot do, and allow the key Stakeholders to make decisions about the quality of the system i.e. is it “good enough”. We use Black Box techniques to prove that the system does what it should do, and doesn’t do what it shouldn’t do, but using Black Box techniques only we don’t know what else has been put into the structure of the system that is outside the defined requirements that may impact on the behaviour of the system. Sometimes Black Box techniques do not allow us to know enough about the system, nor do they provide the level of detail required for testing of higher risk systems.

One of the key approaches to White Box testing is to have access to the detailed design and code and the tools to read the code and assess coverage, or the assistance from the developers in accessing code and assessing the code coverage. We would struggle to do comprehensive System Testing without these White Box approaches. For example, if we had a system that was a Front End Application, which generates a file of input that is then loaded into the database, which then triggers a stored procedure that updates other database tables. Without a White Box test design approach we would not be able to accurately test:

• the contents of the file

• the transfer of the file

• the receipt of the file

• the parsing of the data into the data base

• ensuring that the data is put into the right fields in the right tables in the right formats

• that the stored procedure was kicked off at the right time and did the right activities

5.8.1.2 Black Box Testing

Black box testing is a software testing techniques in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. 7This type of testing is based entirely on the software requirements and specifications.

**Analysis of the input/output domain of the program:**

• Leads to a logical partitioning of the input/output domain into ‘interesting’ subsets

**Analysis of the observable black-box behaviour:**

• Leads to a flow-graph-like model, which enables application of techniques from the white-box world (on the black-box model)

**Heuristics**

• Techniques like risk analysis, random input

5.8.2 Testing & Result

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sr. No | To be test | Specification says | | Actual Result | | Status | |
| 1 | Availability of necessary software and network conditions | 1. Intenet connected Browser | | All requirement available | | Pass | |
| 2 | Configuration Testing | 64 bit operating system   1. Minimum 1 GB RAM 2. Minimum 20MB storage space. | | All requirement available | | Pass | |
| 3 | Compatibility | 1. Bootstrap and MDL Components | | All software’s are compatible | | Pass | |
| 4 | Security | 1. Password must not be readable | | Hashing is implemented to hide password. | | Pass | |
|  |  | | 1. Home page must not be visible after logout | | Session is included to provide this feature. | | Pass | |
| 5 | Error Messages | | 1. Appropriate error messages must be displayed on screen. | | Error messages are appropriately handled. | | Pass | |

**6. System Maintenance**

* 1. **Maintenance**

Website Maintenance comprises all the activities needed to ensure the operational integrity of the website or intranet. In other words, it is about doing all the things needed to make sure our site runs smoothly and according to plan. The activities from which Website Maintenance is composed are:

• Website Publishing: To keep content up-to-date.

• Website Quality Assurance: To spot errors on a site.

• Website Feedback Monitoring: To manage communication with website visitors.

• Website Performance Monitoring: To measure success.

• Change Control: To manage technical and other changes in a coordinated way. These activities are usually carried out by members of a Website.

1. **Future Enhancements**

We would add more features to the system by which we can give more flexibility to the user. We can add different regions of college details module in the upcoming version where user can pay using debit or credit card and will give more flexibility to them. We can also add some more categories of the books so that the scope of the system would increase. Some UI enhancement can also be done in the future.

We can add one more stakeholder to increase the productivity of the system and that’s Suppliers. We would like to implement an algorithm for giving suggestions to the user regarding the type of books they would want to buy.

1. **Limitations**

There are some limitations for the current system to which solutions can be provided as a future development:

Currently, the system only has Mumbai colleges as its data. We plan to include college details of colleges other than Mumbai.

1. **Conclusion**

The Internet has become a major resource in modern business, thus online forums continue to be effective. Hence, analyzing sentiments is very important in the current world. We, hence conclude by saying sentiment analysis will continue to be beneficial in this current world.

1. **Bibliography**
2. https://en.wikipedia.org/wiki/Sentiment\_analysis
3. Dai, Xiangfeng; Spasic, Irena; Andres, Frederic (2017). "A Framework for Automated Rating of Online Reviews Against the Underlying Topics". Proceedings of the SouthEast Conference. ACM SE '17. New York, NY, USA: ACM: 164–167. ISBN 9781450350242. doi:10.1145/3077286.3077291
4. Galitsky, Boris; McKenna, Eugene William. "Sentiment Extraction from Consumer Reviews for Providing Product Recommendations". Retrieved 18 November 2013.
5. Roebuck, K. (2012-10-24). Sentiment Analysis: High-impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors. ISBN 9781743049457.