

PROJECT REPORT

Online Student Feedback System Using Flask

A Report Submitted to

Jawaharlal Nehru Technological University Kakinada, Kakinada

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

Name: KANDALA A V N VAMSI KRISHNA SAI

Regd No: 21KN1A0582



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NRI INSTITUTE OF TECHNOLOGY

Autonomous

(Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)

Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' Grade

ISO 9001: 2015 Certified Institution

Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.

2022-2023



NRI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution, Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada)

Accredited by NBA (CSE, ECE & EEE), Accredited by NAAC with 'A' Grade
ISO 9001: 2015 Certified Institution

Pothavarappadu (V), (Via) Nunna, Agiripalli (M), Krishna Dist., PIN: 521212, A.P, India.

CERTIFICATE

This is to certify that the “**Project report**” submitted by **KANDALA A V N VAMSI KRISHNA (21KN1A0582)**, is work done by him and submitted during 2022-2023 academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING**.

Head of the Department

(Dr. D. SUNEETHA)

ACKNOWLEDGEMENT

We take this opportunity to thank all who have rendered their full support to our work. The pleasure, the achievement, the glory, the satisfaction, the reward, the appreciation and the construction of our project cannot be expressed with a few words for their valuable suggestions.

We are expressing our heartfelt thanks to **Head of the Department, Dr. D. SUNEETHA** garu for her continuous guidance for completion of our Project work.

We are extending our sincere thanks to **Dean of the Department, Dr. K. V. SAMBASIVA RAO** for his continuous guidance and support to complete our project successfully.

We are thankful to the **Principal, Dr. C. NAGA BHASKAR** garu for his encouragement to complete the Project work.

We are extending my sincere and honest thanks to the **Chairman, Dr. R. VENKATA RAO garu & Secretary, Sri K. Sridhar** garu for their continuous support in completing the Project work.

Finally, we thank the Administrative Officer, Staff Members, Faculty of Department of CSE, NRI Institute of Technology and my friends, directly or indirectly helped us in the completion of this project.

Name: KANDALA A V N VAMSI KRISHNA SAI Regd No: 21KN1A0582

INDEX

1. Abstract.....	1
2. Introduction.....	2
2.1 Existing SYSTEM.....	2
2.2 Proposed SYSTEM.....	2-3
2.3 Hardware and Software Requirements.....	3
2.4 Feasibility Study.....	3-6
3. Requirements Specification Document	
3.1 Introduction.....	6
3.2 Functional Requirements.....	6
3.3 Non-Functional Requirements.....	7
3.3.1 Analysis, Design & Data requirements.....	7
3.3.2 Constraints.....	7-8
3.3.3 Guidelines.....	8
3.3.4 Validation criteria.....	8
4 Architecture & Technologies	
4.1 Introduction.....	8-11
4.2 Project Plan.....	12
4.3 System Analysis.....	12-13
4.4 Client / Presentation tier.....	13
4.5 Technologies.....	13-16
5. Design	
5.1 Data Base Design Phase	
5.1.1 Student Data table.....	16
5.1.2 Admin Data table.....	17
5.1.3 Student Feedback table.....	17
5.1.4 Student Comments table.....	17
5.2 Coding Phase.....	18-25
5.3 Frontend Development Phase.....	26-51
5.3 Screen Shots.....	52-59
6. Testing.....	60-61
7. Conclusion.....	61
8. Bibliography.....	61

1. Abstract:

In the realm of education, feedback plays a crucial role in shaping and improving the learning experience. Traditional methods of collecting student feedback have been limited in terms of reach, timeliness, and depth of insights. However, the advent of online technologies has paved the way for the development of an efficient and effective Online Student Feedback System (OSFS). This abstract presents an overview of the key features and benefits of an OSFS, highlighting its potential to enhance education by fostering student engagement, improving teaching practices, and facilitating continuous improvement.

An OSFS provides students with a convenient platform to express their opinions, experiences, and suggestions regarding their courses, instructors, and the overall learning environment. By leveraging online interfaces such as web-based surveys or feedback forms, students can provide feedback at their convenience, eliminating the constraints of physical presence and paper-based surveys. The system enables students to provide anonymous feedback, ensuring a safe space for honest and constructive criticism.

An OSFS promotes student engagement and ownership of the learning process. By actively involving students in providing feedback, they feel valued and recognized as active participants in their education. This fosters a sense of empowerment and encourages a culture of continuous improvement, as students become aware that their opinions are taken seriously and acted upon.

Main Objective:

The primary objective of the Online Student Feedback System (OSFS) is to systematically enhance the educational experience by leveraging technology to gather and utilize student feedback effectively. The OSFS provides a streamlined platform for students to express their opinions and suggestions, thereby encouraging active engagement and fostering a sense of ownership in their education. This system allows instructors to receive timely and constructive feedback, enabling them to refine their teaching methods and better address students' needs. Furthermore, the OSFS promotes a culture of accountability and transparency by demonstrating that student feedback is valued and acted upon to implement meaningful changes. By simplifying the feedback collection process, the OSFS ensures efficiency and accessibility, while the data collected supports informed decision-making and continuous improvement within educational institutions, allowing them to adapt to the evolving demands of students and educators.

Using this software, we can:

- **Submit Feedback:** Students can provide feedback by selecting a course and submitting their opinions and suggestions.
- **Generate Reports:** Educators and administrators can generate reports to analyze feedback trends and identify areas for improvement.
- **Provide Anonymous Feedback:** Students can choose to submit feedback anonymously, ensuring their privacy is protected.

2. Introduction:

An online student feedback system using Python is a powerful tool designed to collect, manage, and analyze feedback from students in an educational setting. This system leverages the versatility and functionality of the Python programming language to create a user-friendly interface for students to provide their feedback on courses, instructors, and overall learning experiences. With features such as secure login, customizable feedback forms, and data visualization capabilities, this system enables educational institutions to gather valuable insights and make informed decisions to enhance the quality of education. By utilizing Python's robust libraries and frameworks, this online student feedback system streamlines the feedback process, promotes student engagement, and facilitates continuous improvement in educational institutions.

CRUD OPERATIONS:

CREATE Operation: Performs the INSERT statement to create a new record.

READ Operation: Reads table records based on the input parameter.

UPDATE Operation: Executes an update statement on the table. It is based on the input parameter.

DELETE Operation: Deletes a specified row in the table. It is also based on the input parameter.

2.1 EXISTING SYSTEM

The existing system for collecting student feedback usually involves manual and paper-based processes. This often includes:

- **Paper Surveys:** Feedback is collected through physical surveys distributed to students in class, which they fill out and return.
- **In-Person Interviews:** Instructors or administrators conduct face-to-face interviews or focus groups with students to gather feedback.
- **Email or Online Forms:** Some institutions may use basic email or generic online forms to collect feedback, lacking integration with other systems.

Disadvantages: Inconvenience. Manual Data Entry. Slow Response Time.

2.2 PROPOSED SYSTEM:

The proposed Online Student Feedback System (OSFS) aims to transform how educational institutions collect, analyze, and utilize student feedback by leveraging modern technology. This web-based platform provides a centralized, user-friendly interface where students can conveniently submit feedback on courses, instructors, and the learning environment from any device with internet access. The system automates feedback collection through online surveys and forms, offering customizable options tailored to specific courses or topics. It ensures anonymity and privacy by allowing students to provide feedback anonymously and securely stores data to protect confidentiality. Real-time data analysis and reporting tools enable quick processing of feedback, generating detailed reports with

visualizations to highlight trends and areas for improvement. Administrators have access to a dashboard for managing submissions and responses. By establishing a feedback, the system supports ongoing enhancement of teaching practices and the learning environment, creating a more dynamic and student-centered educational experience.

2.3 Hardware and Software Requirements

Hardware and software specifications

HARDWARE AND SOFTWARE INTERFACES:

HARDWARE REQUIREMENTS

Client Site:

Processor	: Intel Pentium IV
Speed	: 2.00GHZ
RAM	: 1GB
Hard Disk	: 150 GB
Key Board (104 keys)	: Standard
Screen Resolution	: 1024 x 764 Pixels

Server Site:

Processor	: Intel Pentium IV
Speed	: 2.00GHZ
RAM	: 1GB
Hard Disk	: 150 GB
Key Board (104 keys)	: Standard
Screen Resolution	: 1024 x 764 Pixels

SOFTWARE REQUIREMENTS

OPERATING SYSTEM	: WINDOWS XP AND ABOVE
DATA BASE	: MySQL - mysql-connector-java-8.0.13.jar
PROGRAMMING LANGUAGE	: Python 3
IDE	: Visual Studio Code

2.4 FEASIBILITY STUDY:

Feasibility study is an important phase in the software development process. It enables the developer to have an assessment of the product being developed. It refers to the feasibility study of the product in terms of outcomes of the product, operational use and technical support required for implementing it.

Feasibility study should be performed on the basis of various criteria and parameters. The various feasibility studies are:

- Economic Feasibility
- Operational Feasibility
- Technical Feasibility

Economic Feasibility: It refers to the benefits or outcomes we are deriving from the product compared to the total cost we are spending for developing the product. If the benefits are more or less the same as the older system, then it is not feasible to develop the product.

In the present system, the development of the new product greatly enhances the accuracy of the system and cuts short the delay in the processing of application.

The errors can be greatly reduced and at the same time providing a great level of security. Here we don't need any additional equipment except memory of required capacity. No need for spending money on client for maintenance because the database used is web enabled database.

Operational Feasibility: It refers to the feasibility of the product to be operational. Some products may work very well at design and implementation but may fail in the real time environment. It includes the study of additional human resource required and their technical expertise.

In the present system, all the operations can be performed easily compared to existing system and supports for the backlog data. Hence there is need for additional analysis. It was found that the additional modules added are isolated modules as far as the operational is concerned, so the Developed system is operationally feasible.

Technical Feasibility: It refers to whether the software that is available in the market fully supports the present application. It studies the pros and cons of using particular software for the development and its feasibility. It also studies the additional training needed to be given to the people to make the application work.

In the present system, the user interface is user friendly and does not require much expertise and training. It just needs a mouse click to do any sort of application. The software that is used for developing is server pages fully is highly suitable for the present application since the users require fast access to the web pages and with a high degree of security. This is achieved through integration of web server and database server in the same environment.

Implementation plan:

The main plan for the system developed is to upgrading existing system to the proposed system. There are mainly 4 methods of upgrading the existing system to proposed

- Parallel Run System
- Direct Cut-Over System
- Pilot System
- Phase-in Method

Parallel Run System: It is the most secure method of converting from an existing to new system. In this approach both the systems run in parallel for a specific period of time. During that period if any serious problems were identified while using the new system, the new system is dropped and the older system is taken at the start point again.

Direct Cut -Over Method: In this approach a working version of the system is implemented in one part of the organization such as single work area or department. When the system is deemed complete, it is installed throughout the organization either all at once (direct cut-over) or gradually (phase-in).

Phase-in Method: In this method a part of the system is first implemented and over time other remaining parts are implemented.

Implementation plan used: The workflow Management system is developed on the basis of “Parallel Run Method” because we upgraded the system, which is already in use to fulfill the requirements of the client. The system already in use is treated as the old system and the new system is developed on the basis of the old system and maintained the standards processed by the older system. The upgraded system is working well and is implemented on the client successfully. of the candidate recruitment.

Project Plan

It was decided to use good Software engineering principals in the development of the system since the company had quite a big client network & was aiming to provide staffing for the clients or to develop the internal projects of the companies& expand their operations in the near future. So the following Project Plan was drawn up:

1. The Analysts will interact with the current manual system users to get the Requirements. As a part of this the Requirements Specification Document will be created.
2. The requirements Specifications document will contain the Analysis & Design of the system & ML will be used as the modeling language to express the Analysis & Design of the System. According to Grady Booch et al, in The Unified Modeling Language User Guide [UML-1998], “The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software intensive system. The UML gives you a standard way to write a system's blueprints, covering conceptual things, such as business processes and system functions, as well as concrete things, such as classes written in a specific programming language, database schemas, and reusable software components”.
3. The Analysis, Design, Implementation & testing of the System itself will be broadly based on the Rational Unified Software Development process. According to Ivar Jacobson et al, in The Unified Software Development Process (The Addison-Wesley Object Technology Series) [USDP-2000], the Unified Software Development Process contains Inception, Elaboration, Construction & Transition as the main Phases, which contain further cycles & iterations. This process will be followed to produce an incremental cycle, which will deliver milestones like the Requirements Specification Document etc., at the end of each of the iterations, Phases or cycles.
4. The Architecture & Technologies will be decided as a part of the Analysis of the requirements.
5. Once the Design is ready the Implementation & Testing strategy of the system will commence. Each will be independent of the other. The implementation of the system itself will be broken down into sub-systems following the Software Engineering principles for the development of robust software.
6. Once the implementation is ready, the System testing will take place. If the system is judged to be stable then Acceptance testing by the Users will take place & once the Users are satisfied the System will be rolled out to the Users & they will be trained on how to use it for an initial period.

The following chapters contain an account of how the Technology & architecture for the system were chosen.

3.Requirements Specification Document

3.1 Introduction

According to Roger Pressman in Software Engineering: A Practitioner's Approach (McGraw-Hill Publications) [SEPA–1997], the requirement specification document is produced at the end of Analysis of the system. This document is a very comprehensive document & contains all the User requirements & Analysis diagrams. The Requirements are broadly divided into two groups:

1. Functional requirements
2. Non-functional requirements

3.2 Functional Requirements

The main purpose of functional requirements within the requirement specification document is to define all the activities or operations that take place in the system. These are derived through interactions with the users of the system. Since the Requirements Specification is a comprehensive document & contains a lot of data, it has been broken down into different Chapters in this report. The depiction of the Design of the System in UML is presented in a separate chapter. The Data Dictionary is presented in the Appendix of the system.

1. The System should allows the administrator to manage different levels of tests and their sequence.
2. It allows the administrator to manage the questions in each category.
3. It allows the administrator to manage the list of questions in each category.
4. Candidate can register himself for writing the test.
5. The system then takes the candidate the first level after logging in.
6. The system generates the test by generating questions randomly pickup the questions from the list
7. It allows the candidate to select answers of questions
8. This system finally evaluates the test, display the result and store it.
9. This system can then takes the candidate to the next level.
10. It allows the administrator to generate the report bases on some cut off marks.
11. It allows the candidate the feedback

3.3 Non-Functional Requirements

The non-functional requirements consist of

1. Analysis, Design & Data requirements (Use-case diagrams, textual analysis, sequence diagrams, data dictionary etc.)
2. Constraints.
3. Guidelines.
4. Validation Criteria.

3.3.1 Analysis, Design & Data requirements

The use case diagrams, textual analysis and sequence diagrams & data dictionary fall into this category. Since each category above is of considerable importance, they have been dealt in separate chapters. An outline is only included here.

The Analysis & Design phases of the system yield Use Case diagrams, textual analysis, Sequence Diagrams, Class diagrams & Data Dictionary. Data dictionary consists of process statements showing how data is flowing from starting point to end point.

3.3.2 Constraints

These are the requirements that are not directly related to the functionality of the system. These should be considered as mandatory when the system is developed. The following Constraints were arrived at for the system:

1. The system should be available over the intranet so that the Users like the candidates can use the system from their system which was assigned to him.
2. For gaining entry into the system the users should be registered and should be able use login & passwords for gaining access to the system.
3. The users should be able to change their passwords for increased security.
4. The system should conform to the requirement specified and final deliverables of the project before some date.
5. The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur.
6. There should be no limitation about the hardware platform that is to be used to run the system.
7. Data integrity should be maintained if an error occurs or the whole system comes

down.

8. A user should be registered in the system once in 6 months only.
9. A user can take-up the next level test once he clears previous level.

3.3.3 Guidelines

We have discussed mandatory requirements in the previous section. The requirements in this section should be taken as suggestions & they should be thought of as recommendations to further enhance the usability of the system.

1. The system should display a menu for users to choose from.
2. The system should display users' requests in a reasonable time.
3. Services of the system should be available 24 hours a day.
4. The system should be designed in such a way that it is easy to enhance it with more functionality. It should be scalable & easily maintainable.

3.3.4 Validation Criteria

The Validation Criteria are dealt separately in the Chapter dealing with the Test Strategy & Test cases.

4. Architecture & Technologies

4.1. Introduction

General Methodology in Developing Software Project

The general methodology in developing a system is involved in different phases, which describe the system's life cycle model for developing software project. The concept includes not only forward motion but also have the possibility to return that is cycle back to an activity previously completed. This cycle back or feedback may occur as a result of the failure with the system to meet a performance objective or as a result of changes in redefinition of system activities. Like most systems that life cycle of the computer-based system also exhibits distinct phases.

Those are,

Requirement Analysis Phase

Design Phase

Development Phase

Coding Phase

Testing Phase

1. Requirement Analysis Phase:

This phase includes the identification of the problem, in order to identify the problem; we have to know information about the problem, the purpose of the evaluation for problem to be known. We have to clearly know about the client's requirements and the objectives of the project.

2. Design Phase:

Software design is a process through which the requirements are translated into a representation of software. One of the software requirements have been analyzed and specified, the software design involves three technical activities: design, coding generation and testing. The design of the system is in modular form i.e. the software is logically partitioned into components that perform specific functions and sub functions. The design phase leads to modules that exhibit independent functional characteristics. It even leads to interfaces that reduce the complexity of the connections between modules and with the external environment. The design phase is of main importance because in this activity, decisions ultimately affect the success of software implementation and maintenance.

3. Development Phase:

The development phase includes choosing of suitable software to solve the particular problem given. The various facilities and the sophistication in the selected software give a better development of the problem.

4. Coding Phase:

The coding phase is for translating the design of the system-produced during the design phase into code in a given programming language, which can be executed by a computer and which performs the computation specified by the design.

5. Testing Phase:

Testing is done in various ways such as testing the algorithm, programming code; sample data debugging is also one of following the above testing.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specification are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

Purpose:

The purpose of this document is to describe all external requirements or client provisioning. It also describes the interfaces for the system.

Scope:

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also use by the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alternations without the permission of the client.

System Design:

Design:

Design of software involves conceiving, planning out and specifying the externally observable characteristics of the software product. We have data design, architectural design and user interface design in the design process. These are explained in the following section. The goal of design process is to provide a blue print for implementation, testing and maintenance activities.

The primary activity during data design is to select logical representations of data objects identified during requirement analysis and software analysis. A data dictionary explicitly represents the relationships among data objects and constraints on the elements of the data structure. A data dictionary should be established and used to define both data

and program design.

Design process is in between the analysis and implementation process. The following design diagrams (Data Flow Diagrams and E-R Diagrams) make it easy to understand and implement .

The design process for software system has two levels.

1. System Design or Top-Level Design.
2. Detailed Design or Logical Design.

System Design or Top-Level Design:

In the system design the focus is on deciding which modules are needed for the system, the specification of these modules and how these modules should be interconnected.

Detailed Design or Logical Design:

In detailed design the interconnection of the modules or how the specifications of the modules can be satisfied is decided. Some properties for a software system design are

- Verifiability.
- Completeness.
- Consistency.
- Trace ability.
- Simplicity/Understandability.

The Requirements provided by the users are converted into Users Requirement Specifications described above. The URS documents are then revised, validated, authorized and approved by the users. The development commences after the approval phase i.e. after the signing off of the URS documents. Thus, the URS is concerned to be the most important document from user and developer prospective. The Developer will try to adhere to the requirements specified in the URS documents in order to develop the required application. We have used Waterfall models a development model.

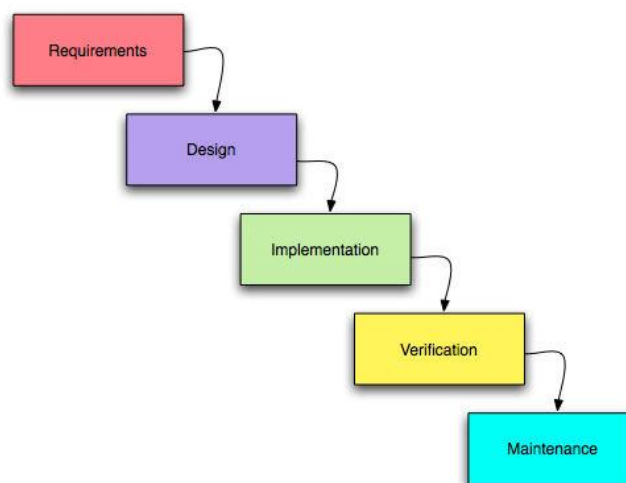


Fig. 2.1 Waterfall Model

4.2. Project Plan

Areas, which would be considered during the planning and analysis, would be:

- Ensure that the information flow is process driven.
- Reduce the manual effort so the maximum extent for all activities.
- Ensure validation at each and every level.
- Act as an effective tool in decision support.
- Provide user friendly system.

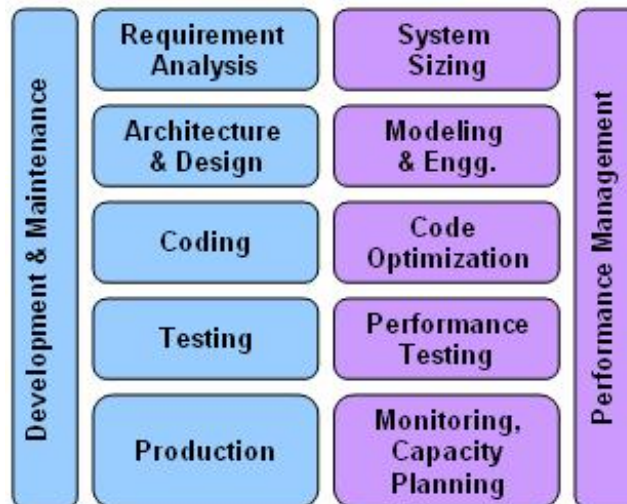


Fig. 2.2 Project Planning and Management Approach

4.3 SYSTEM ANALYSIS:

Analysis is the detailed study of the various operations performed by the system and their relationships within and outside of the system. A key question is: what must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not candidate system should consider other related systems. During analysis, data are collected on the available files, decision points, and transactions handled by the present system.

System Requirement Specification

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of the clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities:

- 1) Problem/Requirement Analysis: The process is order and more nebulous of the two, deals with understanding the problem, the goal and constraints.

- 2) Requirement Specification: Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

Role of SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and the user needs are accurately specified. It forms the basis of the software development. A good SRS should satisfy all the parties involved in the system.

4.4. CLIENT SERVER MODEL

When an architect designs a building, he has a vision of the finished product and produces a result based on that vision. Client – server, on the other hand, is more like Darwinian model of evolution of a living species. No one has a vision of the finished products; rather, day-today events and gradual changes affect it over time in reaction to those events.

In the beginning, application were fairly simple, reading input transaction in a ‘batch’, processing them against a data store, and the output was paper. Record retrieval was usually a set of subroutines embedded in the updating program.

Common functions gradually migrated from the application to the operating system. Database processing was one of the first major functions to be removed from application control. Much of the time database functions in the application included retrieval, replacement and insertion. Since it was function had to be introduced database administration. This new function was separated from the application code and involved defining the structure of the database, value ranges backup, rollback, and so forth.

Advantages of Client – Server Model:

- The hardware and software can be placed where it will do the most good.
- In Client – Server model PCs, the power can be spread across the client and the server.
- On client side, an Active X object is used to present data
- By having the client side it can do more work
- The client software supplies the interface (Such as windowed program) and the knowledge of how to pass the request to the server and the format of the data for the user when it's returned from the server. The server's job is to manipulate the data according to the user's request.

4.5 TECHNOLOGIES

- IDE - Visual Studio Code
- Python 3
- MySQL - mysql-connector-java-8.0.13.jar

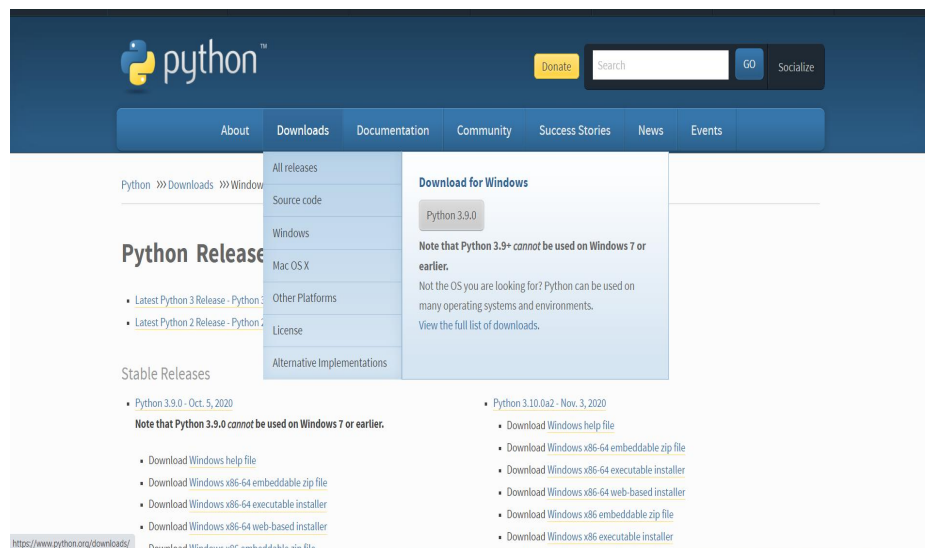
Python:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

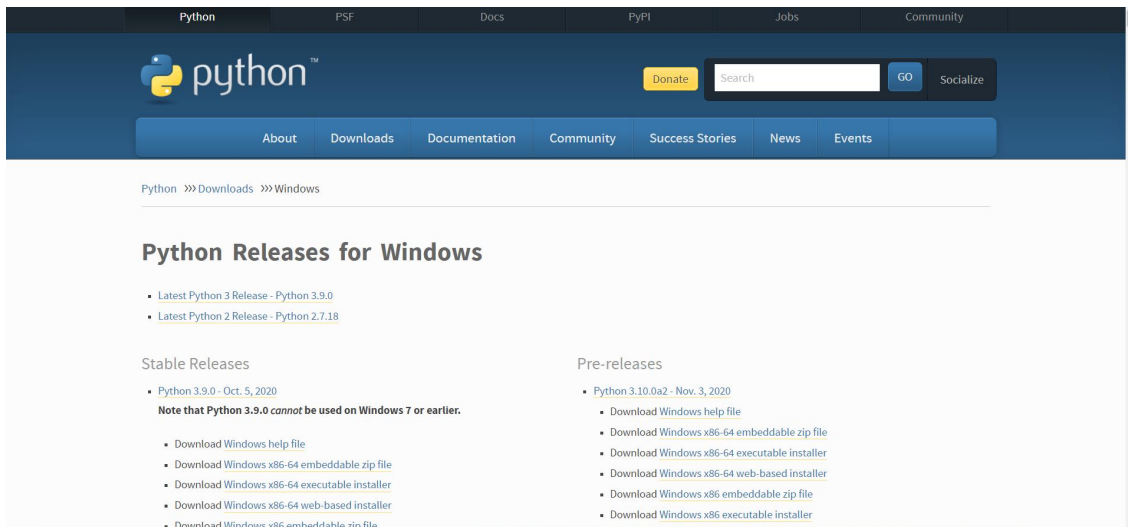


Python Installation and Setting Up Environment:

- Go to the www.python.org and click on the Downloads.
- Select the OS according to your Pc.



- On the next page you will see all the version of python that are available for download.
- Python latest Version is 3.9.0 . Click on the link, It will start the downloading process.



- After downloading, click on run. It will open the installation window.
- Check the box Add Python 3.9 to PATH and then click on install.



MySQL Database:

MySQL is an RDBMS (Relational Database Management System) based on the SQL (Structured Query Language), which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company. The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows. Some of the features of MySQL are listed below:

- It is a database system used on the web
- It runs on a server
- It is ideal for both small and large applications

- It is very fast, reliable, and easy to use
- It uses standard SQL
- It is free to download and use

Python Flask:

Flask is a lightweight Python web framework that provides useful tools and features for creating web applications in the Python Language. It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file.

HTML:

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

5.Designing:

5.1: Data Base Design

The database consists of 2 main tables namely Student data, admin data, and each student has 2 tables namely Comments table and feedback table, each of their structure is given below:

5.1.1 Student Data table:

It consists of all the details that the student should provide.

```
mysql> desc stdata;
```

Field	Type	Null	Key	Default	Extra
rollno	varchar(15)	YES		NULL	
password	varchar(30)	YES		123456	
branch	varchar(10)	YES		NULL	

3 rows in set (0.01 sec)

5.1.1 Student data table

5.1.2 admin table:

It consists of all the details that the admin should provide to access of database.

```
mysql> desc adt;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| username   | varchar(50)   | YES  |     | NULL    |       |
| password   | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5.1.2 admin table

5.1.3 Students Feedback table:

It consists of all the details that the Student Feedback in the database.

```
mysql> desc 21kn1a0582feedback;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| sno        | varchar(100)  | YES  |     | NULL    |       |
| question   | varchar(100)  | YES  |     | NULL    |       |
| os         | varchar(20)   | YES  |     | NULL    |       |
| wt         | varchar(20)   | YES  |     | NULL    |       |
| flat       | varchar(20)   | YES  |     | NULL    |       |
| ps         | varchar(20)   | YES  |     | NULL    |       |
| se         | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

5.1.3 Students Feedback table

5.1.4 Students Comments table:

It consists of all the details that the Student comments in the database.

```
mysql> desc 21kn1a0582comments;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| comments   | varchar(1000) | YES  |     | NULL    |       |
| sno        | varchar(3)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5.1.4 Students Comments table

5.2: Coding:

5.2.1: Importing the required modules and establishing connection with database.

```
from flask import Flask, request, redirect, render_template, session, url_for, flash
import flask_excel as excel
import pyxlsb
import mysql.connector
import secret_key
import io

app = Flask(__name__)
app.secret_key = secret_key
excel.init_excel(app)

mydb=mysql.connector.connect(host="localhost",user="root",password="",db="feedback")
cursor=mydb.cursor()
```

5.2.2: Code for the routing of Home page.

```
@app.route('/')
def home():
    return render_template('home.html')
```

5.2.3: Code for the Student Login.

```
@app.route('/studentlogin', methods=["GET", "POST"])
def studentlogin():
    msg=''
    #Getting the form data
    if request.method == 'POST':
        a1 = request.form['Rollnumber']
        b1 = request.form['password']
        #sql Query to fetch the data according to the user input
        result=cursor.execute("select * from stdata where rollno=%s and password=%s", (a1,b1))
        record = cursor.fetchone()#Fetching the data
        #Checking the user is valid or not. If User is valid then it is redirected to the Student view
        if record:
            session['loggedin']=True
            session['username']=record[0]
            print(session.get('username'))
            return redirect(url_for('studentview'))
        elif not record:
```

```

        return render_template('StudentLogin.html',message='Invalid
Username/Password')

    return render_template('StudentLogin.html')

```

5.2.4: Code for the Student-view.

```

@app.route('/studentview')
def studentview():
    return render_template('student_view.html')

```

5.2.5: Code for the Student Feedback.

```

@app.route('/feedback',methods=['GET','POST'])
def feedback():
    if request.method == 'POST':

        user=str(session.get('username')+'feedback')
        com=session.get('username')+'comments'
        a='os'
        b='wt'
        c='flat'
        d='ps'
        e='se'

        aa1=request.form.get('a1');aa2=request.form.get('a2');aa3=request.form.get('a3');aa4=request.form.get('a4');aa5=request.form.get('a5')
        ba1=request.form.get('b1');ba2=request.form.get('b2');ba3=request.form.get('b3');ba4=request.form.get('b4');ba5=request.form.get('b5')
        ca1=request.form.get('c1');ca2=request.form.get('c2');ca3=request.form.get('c3');ca4=request.form.get('c4');ca5=request.form.get('c5')
        da1=request.form.get('d1');da2=request.form.get('d2');da3=request.form.get('d3');da4=request.form.get('d4');da5=request.form.get('d5')
        ea1=request.form.get('e1');ea2=request.form.get('e2');ea3=request.form.get('e3');ea4=request.form.get('e4');ea5=request.form.get('e5')
        fa1=request.form.get('f1');fa2=request.form.get('f2');fa3=request.form.get('f3');fa4=request.form.get('f4');fa5=request.form.get('f5')
        ga1=request.form.get('g1');ga2=request.form.get('g2');ga3=request.form.get('g3');ga4=request.form.get('g4');ga5=request.form.get('g5')
        ha1=request.form.get('h1');ha2=request.form.get('h2');ha3=request.form.get('h3');ha4=request.form.get('h4');ha5=request.form.get('h5')
        ia1=request.form.get('i1');ia2=request.form.get('i2');ia3=request.form.get('i3');ia4=request.form.get('i4');ia5=request.form.get('i5')
        ja1=request.form.get('j1');ja2=request.form.get('j2');ja3=request.form.get('j3');ja4=request.form.get('j4');ja5=request.form.get('j5')

        print(type(aa1))

```

```

        comment=request.form.get('comments')
        a1=1

        r1=f"update {user} set
{a}='{aa1}',{b}='{aa2}',{c}='{aa3}',{d}='{aa4}',{e}='{aa5}' where sno='1'"
        r2=f"update {user} set
{a}='{ba1}',{b}='{ba2}',{c}='{ba3}',{d}='{ba4}',{e}='{ba5}' where sno='2'"
        r3=f"update {user} set
{a}='{ca1}',{b}='{ca2}',{c}='{ca3}',{d}='{ca4}',{e}='{ca5}' where sno='3'"
        r4=f"update {user} set
{a}='{da1}',{b}='{da2}',{c}='{da3}',{d}='{da4}',{e}='{da5}' where sno='4'"
        r5=f"update {user} set
{a}='{ea1}',{b}='{ea2}',{c}='{ea3}',{d}='{ea4}',{e}='{ea5}' where sno='5'"
        r6=f"update {user} set
{a}='{fa1}',{b}='{fa2}',{c}='{fa3}',{d}='{fa4}',{e}='{fa5}' where sno='6'"
        r7=f"update {user} set
{a}='{ga1}',{b}='{ga2}',{c}='{ga3}',{d}='{ga4}',{e}='{ga5}' where sno='7'"
        r8=f"update {user} set
{a}='{ha1}',{b}='{ha2}',{c}='{ha3}',{d}='{ha4}',{e}='{ha5}' where sno='8'"
        r9=f"update {user} set
{a}='{ia1}',{b}='{ia2}',{c}='{ia3}',{d}='{ia4}',{e}='{ia5}' where sno='9'"
        r10=f"update {user} set
{a}='{ja1}',{b}='{ja2}',{c}='{ja3}',{d}='{ja4}',{e}='{ja5}' where sno='10'"

        result1=cursor.execute(r1)
        result2=cursor.execute(r2)
        result3=cursor.execute(r3)
        result4=cursor.execute(r4)
        result5=cursor.execute(r5)
        result6=cursor.execute(r6)
        result7=cursor.execute(r7)
        result8=cursor.execute(r8)
        result9=cursor.execute(r9)
        result10=cursor.execute(r10)

        commrec=cursor.execute(f"update {com} set comments='{comment}' where
sno='{a1}'")
        mydb.commit()

    return render_template('feedback.html')

```

5.2.6: Code for the Student Logout.

```

@app.route('/studentlogout',methods=["GET","POST"])
def studentlogout():
    if session.get('username'):
        session.pop('username')
        return redirect(url_for('home'))

```


5.2.7: Code For the Admin Login.

```
@app.route('/adminlogin', methods=["GET", "POST"])
def adminlogin():
    msg=''
    #Getting the form data
    if request.method == 'POST':
        ad1 = request.form['username']
        bd1 = request.form['password']
        #sql Query to fetch the data according to the user input
        result=cursor.execute("select * from adt where username=%s and
password=%s",[ad1,bd1])
        record = cursor.fetchone()#Fetching the data
        #Checking the user is valid or not. If User is valid then it is redirected to
the Student view
        if record:
            session['loggedin']=True
            session['username']=record[0]
            session['shf']=''
            session['shfc']=''
            return redirect(url_for('adminview'))
        elif not record:
            return render_template('adminlogin.html',message='Invalid
Username/Password')
        return render_template('adminlogin.html')
```

5.2.8: Code for the Admin-View.

```
@app.route('/Adminview')
def adminview():
    return render_template('adminview.html')

@app.route('/logo')
def logo():
    return render_template('logo.html')

@app.route('/leftmenu')
def leftmenu():
    return render_template('leftmenu.html')

@app.route('/centerbox')
def centerbox():
    return render_template('centerbox.html')

@app.route('/rightmenu')
def rightmenu():
    return render_template('rightmenu.html')
```

5.2.9: Code for the Add Student.

```
@app.route('/addform',methods=["GET", "POST"])
def addform():
    msg=''
    #Getting the form data
    if request.method == 'POST':
        a1 = request.form['rollnumber']
        b1 = request.form['branch']
        c1 = '123456'
        cursor.execute('select count(*) from stdata where rollno=%s',[a1])
        count=cursor.fetchone()[0]
        if count==1:
            cursor.close()
            flash('You are already registerterd!')
            return redirect(url_for('addform'))
        else:
            result=cursor.execute("insert into stdata values(%s,%s,%s)",(a1,c1,b1))
            mydb.commit()
            #sql Query to fetch the data according to the user input
            result=cursor.execute("insert into stdata values(%s,%s,%s)",(a1,c1,b1))

            flash('Student Added Successfully!!')
            mydb.commit()
    return render_template('addform.html',msg=msg)
```

5.2.10: Code for the Update Student Password.

```
@app.route('/updateform',methods=["GET", "POST"])
def updateform():
    msg=''
    #Getting the form data
    if request.method == 'POST':
        a1 = request.form['rollnumber']
        b1 = request.form['branch']
        c1 = request.form['Newpassword']
        cursor.execute('select count(*) from stdata where rollno=%s',[a1])
        count=cursor.fetchone()[0]
        if count==1:
            result=cursor.execute("update stdata set password=%s where rollno=%s",
(c1,a1))
        else:
            flash('No student found with Entered Rollno')
            return redirect(url_for('updateform'))
        mydb.commit()
    return render_template('updateform.html')
```

5.2.11: Code for to Show the Students in Each Department.

```
@app.route('/showstudent',methods=["GET", "POST"])
def showstudent():
    if request.method == 'POST':
        a1 = request.form['branch']
        #sql Query to fetch the data according to the User Input
        result=cursor.execute("select rollno,branch from stdata where
branch=%s",([a1]))
        record=cursor.fetchall()
        return render_template('showstudentdata.html',value=record)

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

5.2.12: Code for to delete the Students in Each Department.

```
@app.route('/deletestudent',methods=["GET", "POST"])
def deletestudent():
    msg=''
    #Getting the form data
    if request.method == 'POST':
        a1 = request.form['rollnumber']
        b1 = request.form['branch']
        #sql Query to fetch the data according to the user input

        result=cursor.execute("delete from stdata where rollno=%s",([a1]))
        mydb.commit()
    return render_template('deletestudent.html')
```

5.2.13: Code for to show feedback the Students in Each Department.

```
@app.route('/showfeedback',methods=["GET", "POST"])
def showfeedback():
    if request.method == 'POST':
        cursor=mydb.cursor(buffered=True)
        roll=request.form['rollnumber']
        a=roll+'feedback'
        b=roll+'comments'
        session['shf']=a
        session['shfc']=b
        result=cursor.execute(f"select
{a}.sno,{a}.question,{a}.os,{a}.wt,{a}.flat,{a}.ps,{a}.se ,{b}.comments from {a} inner
join {b} on {a}.sno={b}.sno")
        record=cursor.fetchall()
        if record:
            return render_template('showfeedbackdata.html',value=record)

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

5.2.14: Code for to reset feedback of the Student.

```
@app.route('/resetfeedback',methods=["GET", "POST"])
def resetfeedback():
    cursor=mydb.cursor(buffered=True)
    a=session.get('shf')
    b=session.get('shfc')
    a1='os'
    b1='wt'
    c='flat'
    d='ps'
    e='se'
    a11='1'
    r1=f"update {a} set {a1}='{''}',
{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='1'"
    r2=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='2'"
    r3=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='3'"
    r4=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='4'"
    r5=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='5'"
    r6=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='6'"
    r7=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='7'"
    r8=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='8'"
    r9=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='9'"
    r10=f"update {a} set
{a1}='{''}' ,{b1}='{''}',{c}='{''}',{d}='{''}',{e}='{''}' where sno='10'"

    result1=cursor.execute(r1)
    result2=cursor.execute(r2)
    result3=cursor.execute(r3)
    result4=cursor.execute(r4)
    result5=cursor.execute(r5)
    result6=cursor.execute(r6)
    result7=cursor.execute(r7)
    result8=cursor.execute(r8)
    result9=cursor.execute(r9)
    result10=cursor.execute(r10)

    commrec=cursor.execute(f"update {b} set comments='{''}' where sno='{a11}'")
    mydb.commit()

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

5.2.15: Code for to add another admin.

```
@app.route('/addadmin',methods=["GET", "POST"])
def addadmin():
    msg=''
    #Getting the form data
    if request.method == 'POST':
        ad1 = request.form['username']
        bd1 = request.form['password']
        #sql Query to fetch the data according to the user input
        result=cursor.execute("insert into adt values(%s,%s)",(ad1,bd1))

        mydb.commit()
    return render_template('addadmin.html')
```

5.2.16: Code for to download the feedback report of the student.

```
@app.route('/download',methods=["GET","POST"])
def download():
    cursor=mydb.cursor(buffered=True)
    a=session.get('shf')
    b=session.get('shfc')
    c='comments'
    d=f"select {c} from {b}"

    cursor.execute(f"select * from {a}")
    data1=cursor.description
    lst=[i[0] for i in data1]
    cursor.execute(d)
    data2=cursor.description
    lst1=[i[0] for i in data2]
    cursor.execute(f"select
{a}.sno,{a}.question,{a}.os,{a}.wt,{a}.flat,{a}.ps,{a}.se ,{b}.comments from {a} inner
join {b} on {a}.sno={b}.sno")
    data=[list(i) for i in cursor.fetchall()]
    data.insert(0,lst+lst1)
    #return (data)
    return excel.make_response_from_array(data, "xls",file_name=f"{a}_report")
```

5.2.17: Code for Admin-Logout.

```
@app.route('/adminlogout',methods=["GET","POST"])
def adminlogout():
    if session.get['username']:
        session.pop('username')
        return redirect(url_for('home'))
    return redirect(url_for('home'))
```

5.3: Frontend Development:

5.3.1: Code for Home Page

CSS:

```
.logo{
  margin:10px auto;
  size: 10%;
  border-radius:15%;
  border-color: bisque;
}

.student_login {
  background-color : red;
  color: white;
  padding: 10px 25px;
  border-radius: 4px;
  border-color: red;
  font-size: 20px;
  font-family: cursive;
}

#student_login_position {
  position: fixed;
  bottom: 10px;
  right: 20px;
}

.admin_login {
  background-color : red;
  color: white;
  padding: 10px 25px;
  border-radius: 4px;
  border-color: red;
  font-size: 20px;
  font-family: cursive;
}

#admin_login_position {
  position: fixed;
  bottom: 10px;
  left: 20px;
}

.admin_login:hover {
  background-color: brown;
  color: white;
}

.student_login:hover {
  background-color: brown;
```

```

    color: white;
}

```

HTML:

```

<html>
  <head>
    <title>NRI</title>
    <style>
      body {

        width:1400;
        height:650;
        animation-name: example;
        animation-duration: 4s;
        animation-iteration-count: infinite;
      }

      @keyframes example {
        0%   {background-color:#2980B9;}
        25%  {background-color: aquamarine,;}
        50%  {background-color: cadetblue;}
        75%  {background-color: #239B56;}
        100% {background-color:#34495E;}
      }

    </style>
    <link rel="stylesheet" href="{{url_for('static',filename='css/home.css')}}">
  </head>
  <body>

    <div class="logo">
      

    </div>
    <div id="student_login_position">
      <a href="{{ url_for('studentlogin') }}"><button class="student_login"
id="student_login" >

         LOGIN</button></a>
      </div>
    <div id="admin_login_position">
      <a href="{{ url_for('adminlogin') }}"><button class="admin_login"
id="adminbutton" >

```

```

         LOGIN</button></a>
    </div>

</body>
</html>

```

5.3.2: Code for the Student Login

CSS

```

.lo{
    background-color:skyblue;
    margin:30px auto;
    padding: 50px;
    color:black;
    font-size: 15px;
    width: 300px;
    height:300px;
    border-radius: 20px;
}
#center{
    margin: 20px auto;
}
#green{
    color:green;
    font-size: 10px;
}
#blue{
    color:blue;
    font-size: 10px;
}
.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}
.submitbutton:hover {
    background-color:lightcoral;
    color: white;
}

```


HTML

```
<html>
  <head>
    <title>
      LOGIN
    </title>
    <style>

      body {

        width:1400;
        height:650;
        animation-name: example;
        animation-duration: 4s;
        animation-iteration-count: infinite;
      }

      @keyframes example {
        0%   {background-color:#2980B9;}
        25%  {background-color: aquamarine,;}
        50%  {background-color: cadetblue;}
        75% {background-color: #239B56;}
        100% {background-color:#34495E;}
      }

    </style>
    <link rel="stylesheet"
href="{{url_for('static',filename='css/studentlogin.css')}}">
  </head>
  <body>
    <div class="logo">
      
    </div>
    <div class="lo" id="divbox">
      <form id="center" method="POST" action="/studentlogin">
        <label>ROLL NUMBER</label>
        <p><input type="text" placeholder="Roll Number" id=RollNumber
name="Rollnumber"></p>
        <label>Password</label>
        <p><input type="password" placeholder="Password" id="Pass"
name="password"></p>
        <p><button class="submitbutton"
id="submitbutton"><div>Submit</button></div></p>
```

```

        </form>
    </div>
    <script>
        {% if message %}
        alert('{{ message }}');
        {% endif %}

    </script>

</body>
</html>

```

5.3.3: Code for the Student View

CSS

```

.feedback{
    background-color : blue;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color: blue;
    font-size: 20px;
    font-family: cursive;
}
#feedback_position {
    position: fixed;
    bottom: 280px;
    right:100px;
    top:630px;
}
.feedback:hover {
    background-color: brown;
    color: white;
}

.admin_login {
    background-color : red;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color: red;
    font-size: 20px;
    font-family: cursive;
}

```

```
#admin_login_position {
    position: fixed;
    bottom: 10px;
    left: 20px;
}
.admin_login:hover {
    background-color: brown;
    color: white;
}
```

HTML

```
<html>
  <head>
    <title>Student View</title>
    <style>
      body {

        width:1400;
        height:650;
        animation-name: example;
        animation-duration: 4s;
        animation-iteration-count: infinite;

      }

      @keyframes example {
        0%   {background-color:#2980B9;}
        25%  {background-color: aquamarine;;}
        50%  {background-color: cadetblue;}
        75%  {background-color: #239B56;}
        100% {background-color:#34495E;}

      }

    </style>
    <link rel="stylesheet"
href="{{url_for('static',filename='css/studentview.css')}}">

  </head>
  <body>
    <div class="logo">
      

    </div>
    <div id="feedback_position">
```

```

        <a href="{ { url_for('feedback') } }"><button class="feedback"
id="feedback">

                 Feedback</button></a>
        </div>

        <div id="admin_login_position">
                <a href="{ { url_for('studentlogout') } }"><button class="admin_login"
id="adminbutton" >

                         LOGOUT</button></a>
                </div>

</body>
</html>

```

5.3.4: Code for the Feedback

CSS

```

.lo{
    background-color:skyblue;
    margin:30px auto;
    padding: 20px;
    color:white;
    font-size: 15px;
    width: 1200px;
    height:784px;
    border-radius: 20px;
}
#center{
    margin: 20px auto;
    padding:5px;
}
th,td{
    padding:15px;
}
.submit_button {
    background-color : red;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color: red;
    font-size: 20px;
    font-family: cursive;
}

```

```

#submit_button_position {
    position: fixed;
    bottom: 10px;
    right: 20px;
}

.submit_button:hover {
    background-color: brown;
    color: white;
}

.commentsdiv{
    border-radius: 5px;
    color:black;
    background-color: white;
    width:700px;
    height:200px;
    margin:15px;
}

textarea{

    margin:10px;
    resize: none;

}

```

HTML

```

<html>
  <head>
    <title>Feed Back</title>
    <style>
      body {

        width:1400;
        height:650;
        animation-name: example;
        animation-duration: 4s;
        animation-iteration-count: infinite;
      }

      @keyframes example {
        0%   {background-color:#2980B9;}
        25%  {background-color: aquamarine,;}
        50%  {background-color: cadetblue;}
        75%  {background-color: #239B56;}
        100% {background-color:#34495E;}
      }
    </style>
  </head>
  <body>

```

```

    }

    </style>
    <link rel="stylesheet"
href="{url_for('static',filename='css/feedback.css')}">
    </head>
    <body width="device-width">
        <div class="logo" >
            

        </div>
        <div class="lo" id="divbox">
            <form id="center" method="post" action="/feedback">
                <table>
                    <th> </th><th> </th><th> OS </th><th>WT </th><th>FLAT </th><th>
P&S</th><th> SE</th>
                    <tr name="one"><td name="1">1</td><td>Is the Faculty Coming to the
Class On time.</td><td><select name="a1"><option>1-Poor</option><option>2-
Neutral</option><option>3-Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td><td><select name="a2"><option>1-
Poor</option><option>2-Neutral</option><option>3-Good</option><option>4-Very
Good</option><option>5-Excellent</option></select></td><td><select
name="a3"><option>1-Poor</option><option>2-Neutral</option><option>3-
Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td><td><select name="a4"><option>1-
Poor</option><option>2-Neutral</option><option>3-Good</option><option>4-Very
Good</option><option>5-Excellent</option></select></td><td><select
name="a5"><option>1-Poor</option><option>2-Neutral</option><option>3-
Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td></tr>
                    <tr name="two"><td>2</td><td>Is the Faculty clearing the
Doubts.</td><td><select name="b1"><option>1-Poor</option><option>2-
Neutral</option><option>3-Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td><td><select name="b2"><option>1-
Poor</option><option>2-Neutral</option><option>3-Good</option><option>4-Very
Good</option><option>5-Excellent</option></select></td><td><select
name="b3"><option>1-Poor</option><option>2-Neutral</option><option>3-
Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td><td><select name="b4"><option>1-
Poor</option><option>2-Neutral</option><option>3-Good</option><option>4-Very
Good</option><option>5-Excellent</option></select></td><td><select
name="b5"><option>1-Poor</option><option>2-Neutral</option><option>3-
Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td></tr>
                    <tr name="three"><td>3</td><td>Is the Faculty Completing the
Syllabus on time.</td><td><select name="c1"><option>1-Poor</option><option>2-
Neutral</option><option>3-Good</option><option>4-Very Good</option><option>5-

```

[illegible]

[illegible]


```

Good</option><option>5-Excellent</option></select></td><td><select
name="j3"><option>1-Poor</option><option>2-Neutral</option><option>3-
Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td><td><select name="j4"><option>1-
Poor</option><option>2-Neutral</option><option>3-Good</option><option>4-Very
Good</option><option>5-Excellent</option></select></td><td><select
name="j5"><option>1-Poor</option><option>2-Neutral</option><option>3-
Good</option><option>4-Very Good</option><option>5-
Excellent</option></select></td></tr>

</table>
<div class="commentsdiv">
    <p><label>Comments</label></p>
    <p><textarea name="comments" padding="10px" rows="8" cols="80"
maxlength="300" ></textarea></p>
</div>
<div id="submit_button_position">
    <button type="submit" class="submit_button" id="submit_button" >

         SUBMIT</button>
    </div>
</form>
</div>
<script>
    {% if message %}
    alert(' {{ message }}')
    {% endif %}
</script>

</body>
</html>

```

5.3.5: Code for the admin login

CSS

```

.lo{
    background-color:skyblue;
    margin:30px auto;
    padding: 50px;
    color:black;
    font-size: 15px;
    width: 300px;
    height:300px;
    border-radius: 20px;
}

```

```
#center{
  margin: 20px auto;
}
#green{
  color:green;
  font-size: 10px;

}
#blue{
  color:blue;
  font-size: 10px;

}
```

HTML

```
<html>
  <head>
    <title>
      ADMIN LOGIN
    </title>
    <style>
      body {

        width:1400;
        height:650;
        animation-name: example;
        animation-duration: 4s;
        animation-iteration-count: infinite;

      }

      @keyframes example {
        0%   {background-color:#2980B9;}
        25%  {background-color: aquamarine,;}
        50%  {background-color: cadetblue;}
        75%  {background-color: #239B56;}
        100% {background-color:#34495E;}

      }
    </style>
    <link rel="stylesheet"
href="{{url_for('static',filename='css/adminlogin.css')}}">
  </head>
  <body>
    <div class="logo">
      

    </div>
```

```

        <div class="lo" id="divbox">
            <form id="center" method="POST" action="/adminlogin">
                <label>USER NAME</label>
                <p><input type="text" placeholder="USERNAME" id=USERNAME
name="username"></p>
                <label>Password</label>
                <p><input type="password" placeholder="Password" id="Pass"
name="password"></p>

                <p><input type="submit" id="SubmitButton"></p>

            </form>
        </div>
        <script>
            {% if message %}
            alert('{{ message }}');
            {% endif %}
        </script>
    </body>
</html>

```

5.3.6: Code for the Admin-view

HTML

```

<html>
<head>
<frameset rows=20,80 frameborder="no">
<frameset cols=*>
<frame src="{{url_for('logo')}}" name=a noresize>
</frameset>
<frameset cols=20,60,20>
<frame src="{{url_for('leftmenu')}}" name=1 noresize>
<frame src="{{url_for('centerbox')}}" name=2 noresize>
<frame src="{{url_for('rightmenu')}}" name=3 noresize>
</frameset>

</head>
<body>
    <div id="admin_login_position">
        <a href="{{ url_for('adminlogout') }}" ><button class="admin_login"
id="adminbutton" >

             LOGOUT</button></a>
    </div>
</body>
</html>

```

5.3.7: Code for the left menu

HTML

```
<html>
<head>
    <link rel="stylesheet" href="{{url_for('static',filename='css/leftmenu.css')}}">

</head>
<body>
<div class="leftmenu" >
    <a href="{{url_for('addform')}}" target="2"><button id="addstudent"
class="button" >Add Student</button></a>
    <a href="{{url_for('updateform')}}" target="2"><button id="updatestudent"
class="button">Update student password</button></a>
    <a href="{{url_for('showstudent')}}" target="2"><button id="showstudent"
class="button" >Show Student</button></a>
    <a href="{{url_for('deletestudent')}}" target="2"><button id="Delete Student"
class="button">Delete Student</button></a>
</div>

</body>
</html>
```

5.3.7: Code for the Right menu

HTML

```
<html>
<head>
    <link rel="stylesheet" href="{{url_for('static',filename='css/rightmenu.css')}}">
</head>
<body>
<div class="rightmenu" >
    <a href="{{url_for('showfeedback')}}" target="2"><button id="showfeedbackresponse"
class="button" >Show Feedback Response</button></a>
    <a href="{{url_for('resetfeedback')}}" target="2"><button
id="Resetfeedbackresponse" class="button" >Reset feedback response</button></a>
    <a href="{{url_for('addadmin')}}" target="2"><button id="addadmin"
class="button" >Add Admin</button></a>
    <a href="{{url_for('download')}}" target="2"><button id="Converttoexcel"
class="button" >Convert to Excel</button></a>
</div>

</body>
</html>
```

5.3.8: Code for the Add Student

CSS

```
.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}
.submitbutton:hover {
    background-color:lightgreen;
    color: white;
}
.addstudent{
    visibility:visible;
    background-color:lightcoral;
    margin:30px auto;
    padding: 50px;
    color:black;
    font-size: 15px;
    width: 300px;
    height:300px;
    border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}
```

HTML

```
<html>
    <head>
        <link rel="stylesheet"
href="{{url_for('static',filename='css/addstudent.css')}}">
    </head>
    <body>
        <div class="addstudent">
            <form method="post" action="">
                <table>Student Rollnumber</table>
                <p><input type="text" placeholder="Roll number" name="rollnumber"></p>
                <label>Branch</label>
                <p><input type="text" placeholder="branch" name="branch"></p>
                <p><button class="submitbutton" id="submitbutton"
onclick="add()"><div>Submit</button></div></p>
            </form>
```

```

</div>
<script>

    function showdata(){
        alert('student added')
    }
</script>
</body>
</html>

```

5.3.9: Code for the Updating of Students Password.

CSS

```

.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}
.submitbutton:hover {
    background-color:lightgreen;
    color: white;
}
.addstudent{
    visibility:visible;
    background-color:lightcoral;
    margin:30px auto;
    padding: 50px;
    color:black;
    font-size: 15px;
    width: 300px;
    height:300px;
    border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}

```

HTML

```

<html>
<head>

```

```

    <link rel="stylesheet"
href="{{url_for('static',filename='css/updatestudentform.css')}}">
</head>
<body>
    <div class="addstudent" id="formdiv">
        <form id="form" method="POST" action="/updateform">
            <lable>Student Rollnumber</lable>
            <p><input type="text" placeholder="Roll number" name="rollnumber"></p>
            <label>Branch</label>
            <p><input type="text" placeholder="branch" name="branch"></p>
            <label>Set New password</label>
            <p><input type="text" placeholder="New password" name="Newpassword"></p>
            <p><button type="submit" class="submitbutton"
id="submitbutton"><div>Update</button></div></p>
            </form>
        </div>

</body>

</html>

```

5.3.10: Code for the Deleting of Students Password.

CSS

```

.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}
.submitbutton:hover {
    background-color:lightgreen;
    color: white;
}
.addstudent{
    visibility:visible;
    background-color:lightcoral;
    margin:30px auto;
    padding: 50px;
    color:black;
    font-size: 15px;
    width: 300px;
    height:300px;
    border-radius: 20px;
}

```

```
body{
background-color:skyblue;
margin:160px 190px;
}
```

HTML

```
<html>
<head>
    <link rel="stylesheet"
href="{url_for('static',filename='css/deletestudentform.css')}">
</head>
<body>
    <div class="addstudent" id="formdiv">
        <form id="form" method="POST" action="/deletestudent">
            <table>Student Rollnumber</table>
            <p><input type="text" placeholder="Roll number" name="rollnumber"></p>
            <label>Branch</label>
            <p><input type="text" placeholder="branch" name="branch"></p>
            <p><button type="submit" class="submitbutton"
id="submitbutton"><div>Delete</div></button></p>
        </form>
    </div>

</body>

</html>
```

5.3.11: Code for the Show Student.

CSS

```
.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}

.submitbutton:hover {
    background-color:lightgreen;
    color: white;
}

.addstudent{
    visibility:visible;
    background-color:lightcoral;
```



```

margin:30px auto;
padding: 50px;
color:black;
font-size: 15px;
width: 300px;
height:300px;
border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}

```

HTML

```

<html>
<head>
  <link rel="stylesheet"
href="{url_for('static',filename='css/showstudentform.css')}">
</head>
<body>
  <div class="addstudent" id="formdiv">
    <form id="form" method="POST" action="/showstudent">
      <label>Branch</label>
      <p><input type="text" placeholder="branch" name="branch"></p>
      <p><button type="submit" class="submitbutton"
id="submitbutton"><div>Show</button></div></p>
    </form>
  </div>
</body>
</html>

```

5.3.12: Code for the Show Student Feedback Form.

CSS

```

.submitbutton {
  background-color :navy;
  color: white;
  padding: 10px 25px;
  border-radius: 4px;
  border-color:lightskyblue;
  font-size: 20px;
  font-family: cursive;
}
.submitbutton:hover {

```

```

background-color:lightgreen;
color: white;
}
.addstudent{
visibility:visible;
background-color:lightcoral;
margin:30px auto;
padding: 50px;
color:black;
font-size: 15px;
width: 300px;
height:300px;
border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}

```

HTML

```

<html>
<head>
  <link rel="stylesheet"
href="{url_for('static',filename='css/showstudentform.css')}">
</head>
<body>
  <div class="addstudent" id="formdiv">
    <form id="form" method="POST" action="/showstudent">
      <label>Branch</label>
      <p><input type="text" placeholder="branch" name="branch"></p>
      <p><button type="submit" class="submitbutton"
id="submitbutton"><div>Show</div></p>
    </form>
  </div>
</body>
</html>

```

5.3.13: Code for the Right Menu.

CSS

```

.rightmenu{
margin:20px 20px;
padding:100px;
}

```

```

background-color:cadetblue;
width:50px;
height:260px;
border-radius:10px;
right:20px;
justify-content: center;
}

button{
left:25px;
border-radius:5px;
width:170px;
height:50px;
background-color:blue;
color:white;
margin:10px -55px;
}
body {
    animation-name: example;
    animation-duration: 4s;
    animation-iteration-count: infinite;
}
@keyframes example {
    0% {background-color:#2980B9;}
    25% {background-color: aquamarine,;}
    50% {background-color: cadetblue;}
    75% {background-color: #239B56;}
    100% {background-color:#34495E;}
}
button:hover{
background-color:lightpink;
}

```

HTML

```

<html>
<head>
    <link rel="stylesheet" href="{{url_for('static',filename='css/rightmenu.css')}}">
</head>
<body>
<div class="rightmenu" >
    <a href="{{url_for('showfeedback')}}" target="2"><button id="showfeedbackresponse"
class="button" >Show Feedback Response</button></a>
    <a href="{{url_for('resetfeedback')}}" target="2"><button
id="Resetfeedbackresponse" class="button" >Reset feedback response</button></a>

```

```

    <a href="{{url_for('addadmin')}}" target="2"><button id="addadmin"
class="button" >Add Admin</button></a>
    <a href="{{url_for('download')}}" target="2"><button id="Converttoexcel"
class="button" >Convert to Excel</button></a>
</div>

</body>
</html>

```

5.3.14: Code for the Show Student Feedback Response.

CSS

```

.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}
.submitbutton:hover {
    background-color:lightgreen;
    color: white;
}
.addstudent{
    visibility:visible;
    background-color:lightcoral;
    margin:30px auto;
    padding: 50px;
    color:black;
    font-size: 15px;
    width: 300px;
    height:300px;
    border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}

```

HTML

```

<html>
    <head>
        <link rel="stylesheet"
href="{{url_for('static',filename='css/showstudentform.css')}}">
    </head>
    <table border="1" align="left" padding="5px" cellspacing="8px" cellpadding="10px">

```

```

<thead>
<tr>
    <th>SNO</th>
    <th>Question</th>
    <th>OS</th>
    <th>WT</th>
    <th>FLAT</th>
    <th>P&S</th>
    <th>SE</th>
    <th>Comments</th>
</tr>
</thead>
<tbody>
    {% for row in value %}
        <tr>
            <td>{{row[0]}}</td>
            <td>{{row[1]}}</td>
            <td>{{row[2]}}</td>
            <td>{{row[3]}}</td>
            <td>{{row[4]}}</td>
            <td>{{row[5]}}</td>
            <td>{{row[6]}}</td>
            <td>{{row[7]}}</td>
        </tr>

    {% endfor %}

</tbody>
</table>
</html>

```

5.3.15: Code for the Reset Student Feedback Response.

CSS

```

.submitbutton {
    background-color :navy;
    color: white;
    padding: 10px 25px;
    border-radius: 4px;
    border-color:lightskyblue;
    font-size: 20px;
    font-family: cursive;
}
.submitbutton:hover {
    background-color:lightgreen;
    color: white;
}
.addstudent{
    visibility:visible;
}

```

```

background-color:lightcoral;
margin:30px auto;
padding: 50px;
color:black;
font-size: 15px;
width: 300px;
height:300px;
border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}

```

HTML

```

<html>
  <head>
    <link rel="stylesheet"
href="{url_for('static',filename='css/showstudentform.css')}">
  </head>
  <body>
  </body>
</html>

```

5.3.16: Code for the adding of another admin.

CSS

```

.submitbutton {
background-color :navy;
color: white;
padding: 10px 25px;
border-radius: 4px;
border-color:lightskyblue;
font-size: 20px;
font-family: cursive;
}
.submitbutton:hover {
background-color:lightgreen;
color: white;
}
.addstudent{
visibility:visible;
background-color:lightcoral;
margin:30px auto;
padding: 50px;
color:black;
font-size: 15px;
width: 300px;
}

```

```

    height:300px;
    border-radius: 20px;
}
body{
background-color:skyblue;
margin:160px 190px;
}

```

HTML

```

<html>
<head>
    <link rel="stylesheet" href="{url_for('static',filename='css/addstudent.css')}">
</head>
<body>
    <div class="addstudent" id="formdiv">
        <form id="form" method="POST" action="/addadmin">
            <label>Username</label>
            <p><input type="text" placeholder="username" name="username"></p>
            <label>Password</label>
            <p><input type="text" placeholder="password" name="password"></p>
            <p><button type="submit" class="submitbutton"
id="submitbutton"><div>Add</button></div></p>
            </form>
        </div>
    </body>
</html>

```

5.3.17: Code for the Download the Student Feedback Report.

HTML

```

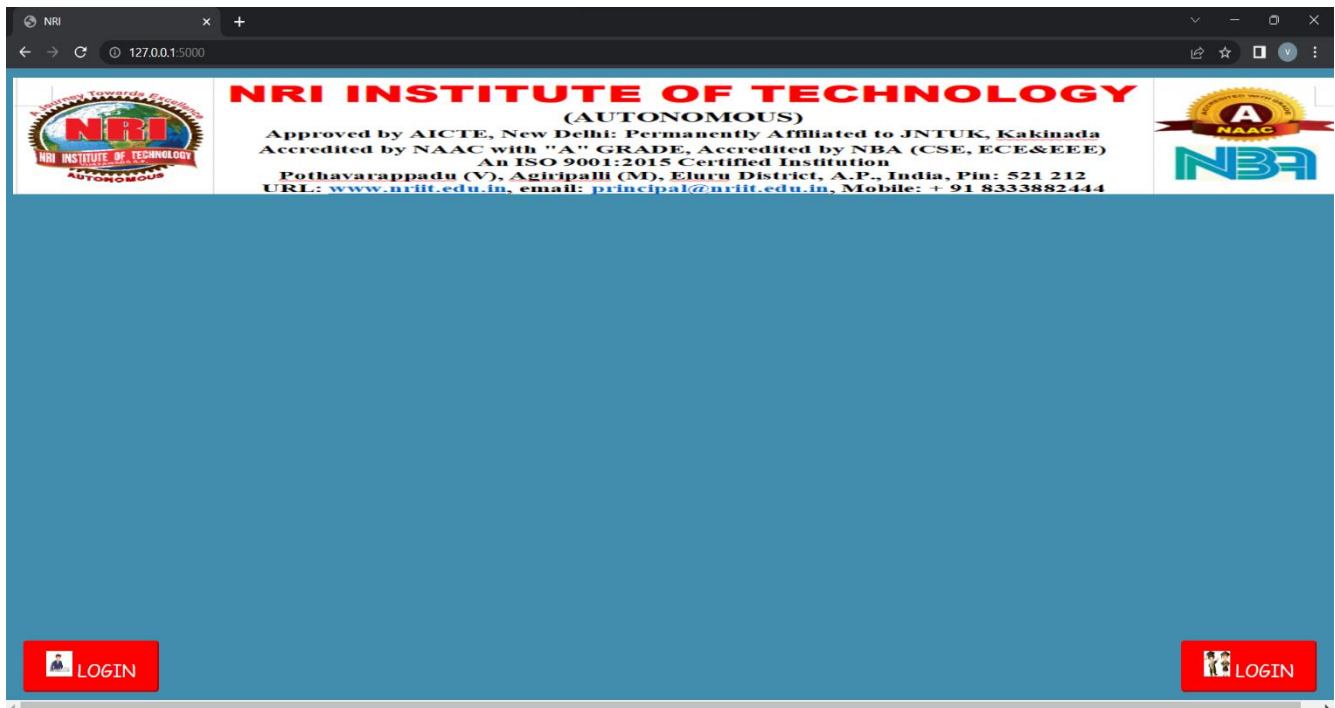
<html>
    <head>
        <link rel="stylesheet"
href="{url_for('static',filename='css/showstudentform.css')}">
    </head>
    <body>
        <h2>Generate Excel Report from FeedBack data</h2>
        <p>
            <a href="{ url_for('download') }">Generate Excel Report</a>
        </p>
    </body>
</html>

```

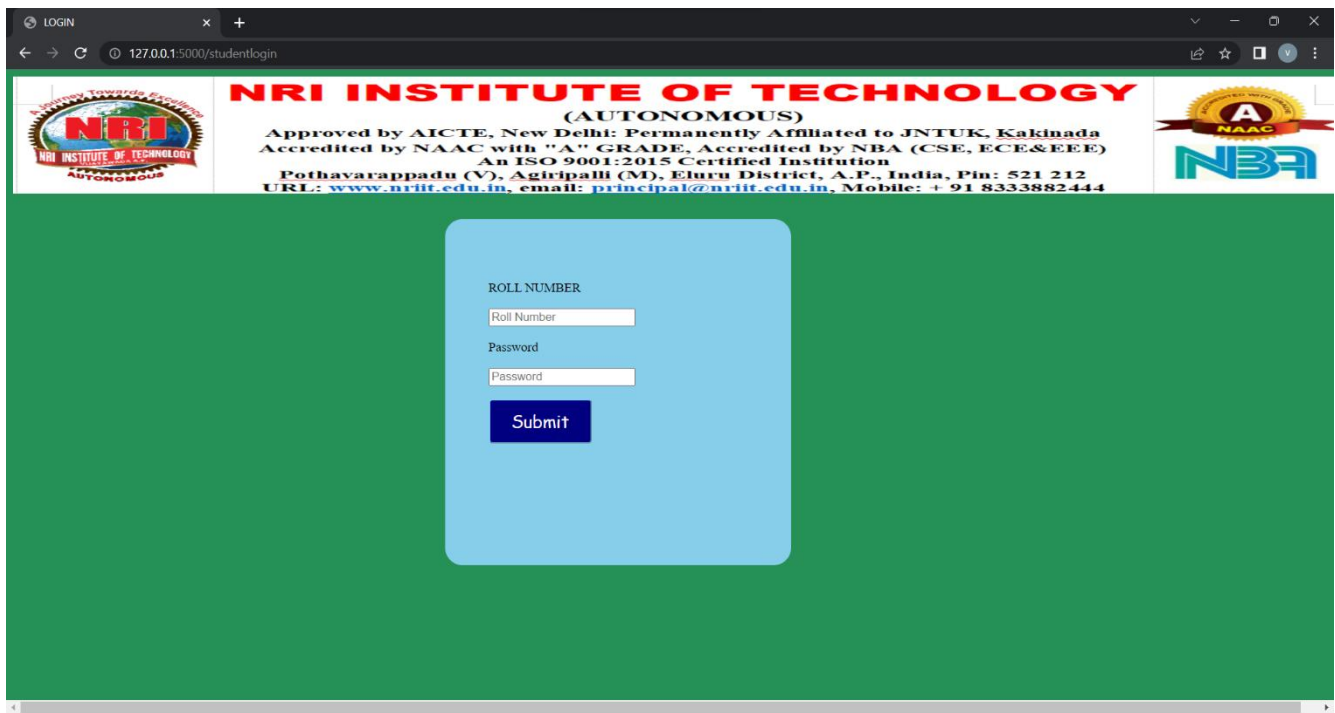
5.4: Screen Shots:

User Side:

HOME



Student Login



Student View

NRI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
An ISO 9001:2015 Certified Institution
Pothavarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
URL: www.nriit.edu.in, email: principal@nriit.edu.in, Mobile: + 91 8333882444

LOGOUT **Feedback**

Student Feedback Form

NRI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
An ISO 9001:2015 Certified Institution
Pothavarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
URL: www.nriit.edu.in, email: principal@nriit.edu.in, Mobile: + 91 8333882444

	OS	WT	FLAT	P&S	SE
1 Is the Faculty Coming to the Class On time.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
2 Is the Faculty clearing the Doubts.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
3 Is the Faculty Completing the Syllabus on time.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
4 Is the Faculty Encourage the students to participate in the seminar.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
5 Is the Faculty giving Examples.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
6 Is the Faculty Giving Tasks to understand the topics.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
7 Is the Faculty Finding the weakness and gives the support.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
8 Is the Faculty teaching in the Practical Oriented.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
9 Is the Faculty being the Mentor and Giving the guidance.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor
10 Is the Faculty Eco Friendly.	1-Poor	1-Poor	1-Poor	1-Poor	1-Poor

Comments
All good

SUBMIT

Admin Login

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/adminlogin". The page header features the NRI Institute of Technology logo on the left, the institute's name and accreditation details in the center, and the NAAC and NBA logos on the right. The main content area has a green background with a light blue login box in the center. The login box contains the following fields and buttons:

USER NAME

Password

Admin View

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/Adminview". The page header is identical to the Admin Login page. The main content area has a light blue background with a large central white box. On the left and right sides of this central box are two vertical panels, each containing four blue buttons:

Left Panel:

- Add Student
- Update student password
- Show Student
- Delete Student

Right Panel:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

Add Student

The screenshot shows a web browser window with the URL 127.0.0.1:5000/Adminview. The page header features the NRI Institute of Technology logo on the left, the institute's name and accreditation details in the center, and the NAAC and NBA logos on the right. The main content area has a light blue background. On the left side, there is a vertical sidebar with four blue buttons: 'Add Student', 'Update student password', 'Show Student', and 'Delete Student'. On the right side, there is another vertical sidebar with four blue buttons: 'Show Feedback Response', 'Reset feedback response', 'Add Admin', and 'Convert to Excel'. In the center, there is a red rectangular form titled 'Student Rollnumber'. It contains two input fields: 'Roll number' and 'Branch', followed by a blue 'Submit' button.

Update Student Password

The screenshot shows the same web browser window as the previous one, but the central form is now titled 'Set New password'. It contains two input fields: 'New password' and 'Confirm password', followed by a blue 'Update' button. The rest of the page layout, including the header, sidebar buttons, and logos, remains identical to the previous screenshot.

Show Student

127.0.0.1:5000/Adminview x +
127.0.0.1:5000/Adminview

NRI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
An ISO 9001:2015 Certified Institution
Pothayarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
URL: www.nrilit.edu.in, email: principal@nrilit.edu.in, Mobile: + 91 8333882444

Left Sidebar:

- Add Student
- Update student password
- Show Student
- Delete Student

Center Form:

Branch

Show

Right Sidebar:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

127.0.0.1:5000/Adminview x +
127.0.0.1:5000/Adminview

NRI INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
An ISO 9001:2015 Certified Institution
Pothayarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
URL: www.nrilit.edu.in, email: principal@nrilit.edu.in, Mobile: + 91 8333882444

Left Sidebar:

- Add Student
- Update student password
- Show Student
- Delete Student

Center Table:

Roll Number	Branch
21KN1A0582	CSE
21KN1A05B4	CSE
21KN1A0579	CSE

Right Sidebar:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

Delete Student

The screenshot shows a web browser window with the URL 127.0.0.1:5000/Adminview. The page header includes the NRI Institute of Technology logo, the text "NRI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)", and accreditation details: "Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada Accredited by NAAC with 'A' GRADE, Accredited by NBA (CSE, ECE&EEE) An ISO 9001:2015 Certified Institution". The address is "Pothayarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212" and the URL is "www.nrilt.edu.in".

The main content area has a light blue background. On the left, there is a vertical sidebar with four blue buttons: "Add Student", "Update student password", "Show Student", and "Delete Student". On the right, there is another vertical sidebar with four blue buttons: "Show Feedback Response", "Reset feedback response", "Add Admin", and "Convert to Excel".

In the center, there is a red rectangular form titled "Delete Student". It contains two input fields: "Student Rollnumber" with the value "21KN1A05C0" and "Branch" with the value "CSE". Below these fields is a blue button labeled "Delete".

Show Student Feedback Response

The screenshot shows the same web browser window as the previous one, but the central form is now titled "Show Student Feedback Response". It contains one input field labeled "Enter Student Rollnumber to Show feedback" with the value "21KN1A0582". Below this field is a blue button labeled "Submit".

The sidebars and header information remain the same as in the previous screenshot.

Reset Student Feedback Response

127.0.0.1:5000/Adminview

127.0.0.1:5000/Adminview

NRI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)
 Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada
 Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
 An ISO 9001:2015 Certified Institution
 Pothavaram (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
 URL: www.nrilit.edu.in, email: principal@nrilit.edu.in, Mobile: + 91 8333882444

Left Sidebar:

- Add Student
- Update student password
- Show Student
- Delete Student

SNO	Question	OS	WT	FLAT	P&S	SE	Comments
1	Is the Faculty Coming to the Class On time.	1- Poor	1- Poor	1- Poor	1- Poor	1- Poor	All good
2	Is the Faculty clearing the Doubts.	1- Poor	1- Poor	1- Poor	1- Poor	1- Poor	
3	Is the Faculty Completing the Syllabus on time.	1- Poor	1- Poor	1- Poor	1- Poor	1- Poor	

Right Sidebar:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

127.0.0.1:5000/resetfeedback

127.0.0.1:5000/Adminview

127.0.0.1:5000/Adminview

NRI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)
 Approved by AICTE, New Delhi; Permanently Affiliated to JNTUK, Kakinada
 Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
 An ISO 9001:2015 Certified Institution
 Pothavaram (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
 URL: www.nrilit.edu.in, email: principal@nrilit.edu.in, Mobile: + 91 8333882444

Left Sidebar:

- Add Student
- Update student password
- Show Student
- Delete Student

SNO	Question	OS	WT	FLAT	P&S	SE	Comments
1	Is the Faculty Coming to the Class On time.						
2	Is the Faculty clearing the Doubts.						
3	Is the Faculty Completing the Syllabus on time.						
4	Is the Faculty Encourage the students to participate in the seminar.						
5	Is the Faculty giving Examples.						
6	Is the Faculty Giving Tasks to understand the topics.						
7	Is the Faculty Finding the weakness and gives the support.						
8	Is the Faculty teaching in the Practical Oriented.						
9	Is the Faculty being the Mentor and Giving the guidance.						
10	Is the Faculty Eco Friendly.						

Right Sidebar:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

Add Admin

The screenshot shows the 'Add Admin' page of the NRI Institute of Technology Adminview. The page has a header with the institute's name and accreditation details. The main content area features a central form for adding a new admin user. The form includes fields for 'Username' and 'Password', and an 'Add' button. On the left side, there are buttons for 'Add Student', 'Update student password', 'Show Student', and 'Delete Student'. On the right side, there are buttons for 'Show Feedback Response', 'Reset feedback response', 'Add Admin', and 'Convert to Excel'.

NRI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)
 Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
 Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
 An ISO 9001:2015 Certified Institution
 Pothavarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
 URL: www.nrilit.edu.in, email: principal@nrilit.edu.in, Mobile: + 91 8333882444

Form Fields:

- Username:
- Password:
- Add**

Left Sidebar Buttons:

- Add Student
- Update student password
- Show Student
- Delete Student

Right Sidebar Buttons:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

Convert to Excel

The screenshot shows the 'Convert to Excel' page of the NRI Institute of Technology Adminview. The page has a header with the institute's name and accreditation details. The main content area features a table with feedback data. The table has columns for SNO, Question, OS, WT, FLAT, P&S, SE, and Comments. The table contains three rows of data. On the left side, there are buttons for 'Add Student', 'Update student password', 'Show Student', and 'Delete Student'. On the right side, there are buttons for 'Show Feedback Response', 'Reset feedback response', 'Add Admin', and 'Convert to Excel'.

NRI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)
 Approved by AICTE, New Delhi: Permanently Affiliated to JNTUK, Kakinada
 Accredited by NAAC with "A" GRADE, Accredited by NBA (CSE, ECE&EEE)
 An ISO 9001:2015 Certified Institution
 Pothavarappadu (V), Agiripalli (M), Eluru District, A.P., India, Pin: 521 212
 URL: www.nrilit.edu.in, email: principal@nrilit.edu.in, Mobile: + 91 8333882444

Table Data:

SNO	Question	OS	WT	FLAT	P&S	SE	Comments
1	Is the Faculty Coming to the Class On time.	1- Poor	1- Poor	1- Poor	1- Poor	1- Poor	All good
2	Is the Faculty clearing the Doubts.	1- Poor	1- Poor	1- Poor	1- Poor	1- Poor	
3	Is the Faculty Completing the Syllabus on	1- Poor	1- Poor	1- Poor	1- Poor	1- Poor	

Left Sidebar Buttons:

- Add Student
- Update student password
- Show Student
- Delete Student

Right Sidebar Buttons:

- Show Feedback Response
- Reset feedback response
- Add Admin
- Convert to Excel

6.TESTING

Software testing is a critical element of the software quality assurance and represents the ultimate review of specification, design and coding. Testing is the exposure of the system to trial input to see whether it produces correct output.

Testing Phases: Software testing phases include the following:

- Test activities are determined and test data selected.
- The test is conducted and test results are compared with the expected results.

There are various types of testing:

Unit Testing: Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

Integration Testing: All the tested modules are combined into subsystems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. The generic code integration testing is done mainly on table creation module and insertion module.

System Testing: It is mainly used if the software meets its requirements. The reference document for this process is the requirement document. **Acceptance Testing:** It is performed with realistic data of the client to demonstrate that the software is working satisfactorily.

Testing Methods: Testing is a process of executing a program to find out errors. If testing is conducted successfully, it will uncover all the errors in the software. Any testing can be done basing on two ways:

White Box Testing: It is a test case design method that uses the control structures of the procedural design to derive the test cases. Using this testing a software engineer can derive the following test cases: Exercise all the logical decisions on either true or false sides. Execute all loops at their boundaries and within their operational boundaries. Exercise the internal data structures to assure their validity.

Black Box Testing: It is a test case design method used on the functional requirements of the software. It will help a software engineer to derive sets of input conditions that will exercise all the functional requirements of the program.

Black box Testing attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures
- Performance errors
- Initialization and termination errors

By Black box testing we derive a set of test cases that satisfy the following criteria:

- Test cases that reduce by a count that is greater than one, the number of additional test cases that must be designed to achieve reasonable testing.
- Test cases that tell us something about the presence or absence of classes of errors rather than errors associated only with a specific test at hand.

TEST APPROACH:

Testing can be done in two ways:

- Bottom up approach
- Top down approach

Bottom up approach: Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

Top down approach: This type of testing starts from upper level modules, since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

7. Conclusion:

The Web Application Using Python Flask for the registering complaints using CRUD Operation with the Connection to Database Using MySQL was successful in creating a secure and user-friendly experience for users. The documentation provides a comprehensive overview of the project and can be used as a reference for future development or maintenance.

8. BIBLIOGRAPHY

WEBSITES REFERRED:

<https://www.python.org/>
[https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
<https://www.learnpython.org/>
<https://realpython.com/>
<https://flask.palletsprojects.com/en/2.3.x/>
<https://www.tutorialspoint.com/flask/index.htm>
<https://www.fullstackpython.com/flask.html>