VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS

An Undergraduate Thesis Manuscript
Submitted to the Faculty of the
Department of Computer Studies of
Cavite State University - CCAT Campus
Rosario, Cavite

In partial fulfilment
of the requirements for the degree,
Bachelor of Science in Computer Science

ELGENE JOHN V. REYES
KARL DOHNEYZEL A. DELACRUZ
JAN PAOLO A. CANO
January 2024

APPROVAL SHEET

Republic of the Philippines
CAVITE STATE UNIVERSITY
CCAT Campus
Rosario, Cavite
(046) 437-9505 / (046) 437-6659
cvsurosario@cvsu.edu.ph
www.cvsu-rosario.edu.ph

DEPARTMENT OF COMPUTER STUDIES

Name:	:	**ELGENE JOHN V. REYES** **KARL DOHNEYZEL A. DELACRUZ** **JAN PAOLO A. CANO**
Degree//Specialization	:	**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**
Title	:	**VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS**

CHRISTOPHER G. ESTONILO, DIT 2/8/2024
Adviser — Date

JEFFRID D. LAXAMANA 2/8/2024
Technical Critic — Date

JENERRY Y. ABAD 2/9/2024
Unit Research Coordinator — Date

KARLO JOSE E. NABABLIT, MSICT 02/08/24
Department Chairperson — Date

ARIES M. GELERA, MSICT 02/12/24
Director, Research and Extension — Date

ARIEL G. SANTOS, EdD
Director, Instruction — Date

LAURO B. PASCUA, EdD 02.12.24
Campus Administrator — Date

Form Number	Revision Date	Effectivity Date
CvSU-CCAT SR Form No. 6	November 21, 2022	November 25, 2022

BIOGRAPHICAL DATA

Elgene John V. Reyes was born on September 15, 2000, in General Trias, Cavite. He is the eldest of two children of Mr. Edgardo Reyes Jr. and Ms. Judylane Villaflor and currently resides in Tanza, Cavite.

He completed his elementary education with honors at Tejeros Convention Elementary School in Rosario, Cavite, 2013 and finished junior high school at Cavite State University-CCAT Campus in Rosario, Cavite, in 2017.

For his senior high school, he attended St. Mary Magdalene School in Kawit, Cavite, where he graduated with high honors in 2019.

He pursued a Bachelor of Science in Computer Science and graduated from Cavite State University-CCAT Campus in 2024.

BIOGRAPHICAL DATA

Karl Dohneyzel A. Dela Cruz was born on November 18, 2000 in Bacoor City, Cavite. He is the eldest of five children of Mr. Dahnyzel A. Dela Cruz and Ms. Jesheryl T. Abodeso, living in Bacoor City, Cavite.

He completed his Elementary education at Sineguelasan Elementary School in Bacoor, Cavite, in 2013. He then attended Junior High School at Bacoor National High School-Tabing Dagat in Bacoor, Cavite, graduating in 2017. In 2019, he completed his Senior High School at City of Bacoor Senior High School-Dulong Bayan, located within Sineguelasan Elementary School, with honors.

He earned his Bachelor of Science in Computer Science from Cavite State University-CCAT Campus in 2024.

BIOGRAPHICAL DATA

Jan Paolo A. Cano was born in Tondo, Manila, on January 16, 2001. He is the second child of Mr. Ruben A. Cano Jr. and Mrs. Catrina A. Cano. Currently, he resides in General Trias, Cavite.

In 2013, he completed his elementary education at David P. Jimenez Elementary School in Rosario, Cavite. He graduated with honors from Bagbag National High School in Rosario, Cavite, in 2017, completing his junior high school education. His senior high school education was completed at Rosario Institute in Rosario, Cavite, in 2019.

In 2024, he earned his Bachelor of Science degree in Computer Science from Cavite State University-CCAT Campus.

ABSTRACT

ELGENE JOHN V. REYES, KARL DOHNEYZEL A. DELACRUZ, JAN PAOLO A. CANO. VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS. An Undergraduate Thesis Manuscript. Department of Computer Studies. Cavite State University-CCAT Campus, Rosario, Cavite. January 2024. Adviser: Dr. Christopher G. Estonilo.

The VisiThoughts web application was developed for the Cavite State University-CCAT Campus (CvSU-CCAT Campus) to analyze customer comments using sentiment analysis. The application proved the efficient implementation of the specialized Natural Language Toolkit (NLTK) algorithm. This advanced system was a great help to visitors, department heads, and administrators because it made it easy to collect detailed input from several offices on the CvSU-CCAT Campus. The research spanned from June 2023 to January 2024 at the CvSU-CCAT Campus. It was divided into four phases: 1) identified the essential requirements and functionalities of the web application through the following: a) conducted a thorough analysis of the feedback collection method and analysis needs of various offices and units within Cavite State University-CCAT Campus; b) identified the essential features and functionalities that should be incorporated into the VisiThoughts web application to meet the identified needs; and c) defined the user interface design and user experience guidelines to ensure a user-friendly and accessible platform; 2) developed the VisiThoughts web application using Lexicon-based approach; 3) evaluated the application in order to determine if it complied with the modified ISO-IEC 25010 standards; and 4) implemented the web application in CvSU-CCAT Campus.

The evaluation of the system involved 31 participants, including five (5) IT experts and twenty-six end users. The results of the assessment revealed an impressive average score of 4.88 from the IT experts, characterized as ‘Excellent’, and an average score of 4.69 from the end users, also interpreted as ‘Excellent’.

ACKNOWLEDGMENT

This research resulted from the unwavering support and direction of many people who made significant contributions to the planning and execution of this study. The following people have the authors' sincere gratitude and appreciation:

Above all, we thank God Almighty for providing us with thoughts, understanding, and discernment, which have helped us overcome the many theories, concepts, and obstacles we have come across in our studies.

Dr. Lauro B. Pascua, campus administrator, for his support and inspiration.

Dr. Ariel G. Santos, director of instructors, for his advice and assistance.

Prof. Aries M. Gelera, director for research and extension, and Prof Karlo Jose E. Nabablit, the current department chairperson, thank you for all of your guidance and support during this study and for your constant encouragement.

Dr. Christopher G. Estonilo, thesis adviser, for his unwavering patience and direction as we developed our concepts and systems, which enabled us to conduct significant research.

Mr. Jeffrid D. Laxamana, technical critic, for his invaluable time, ideas, and technical advice in helping us improve our system and

To their loving parents, who have always supported and helped them throughout this journey. This modest endeavor is dedicated to each of them with heartfelt gratitude.

EJVR, KDADC, JPAC

TABLE OF CONTENTS

APPROVAL SHEET ..... **ii**

BIOGRAPHICAL DATA ..... **iii**

ABSTRACT ..... **vi**

ACKNOWLEDGMENT ..... **viii**

TABLE OF CONTENTS ..... **ix**

LIST OF TABLES ..... **xii**

LIST OF APPENDICES ..... **xiii**

LIST OF FIGURES ..... **xiv**

INTRODUCTION **1**

1.1 Project Context 1

1.2 Purpose and Description 4

1.3 Objectives of the Study 5

1.4 Scope and Limitations 6

1.5 Concept of the Study 8

1.6 Definition of Terms 9

2 REVIEW OF RELATED LITERATURE **11**

2.1 Importance of Feedback 11

2.2 Sentiment Analysis in Interpreting Feedback 12

2.3 Lexicon-based Approach for Sentiment Analysis 13

2.4 Effectiveness of Web Application for Collecting Feedback 14

2.5 Python in sentiments analysis 15

2.6 Bootstrap, HTML, CSS, and Java Script for creating web application 16

2.7 ISO EIC 25010 17

2.8 SQLite as web server for web applications 18

2.9 Importance and benefits of using Django in creating web application with sentiment analysis 19

2.10 Automating Large Scale Healthcare Service Feedback Analysis: Using Natural Language Processing and Topic Modelling (Preprint) 20

2.11 Opinion mining technique for developing student feedback analysis system using lexicon-based approach (OMFeedback) 20

2.12 Web-Based Student Opinion Mining System Using Sentiment Analysis ... 21

2.13 Sentiment Analysis for University Students' Feedback 21

2.14 Sentiresearch: Lexicon- based web application for indonesian sentiment analysis 22

2.15 Synthesis 23

3 MATERIALS AND METHODOLOGY 25

3.1 Materials 25

3.1.1 Software 25

3.1.2 Hardware 26

3.1.3 Data 26

3.2 Methodology 26

3.2.1 Data Analysis 27

3.2.2 Development of the Application 27

3.2.3 Evaluation of the Application 30

3.2.4 Deployment Plan 36

4 RESULTS AND DISCUSSION 37

4.1 Results of the data analysis and model building 37

4.2 Results of the application development 40

4.3 Results of the application evaluation 49

4.3.1 Functional Testing 49

4.3.2 Software Evaluation from the End Users 51

4.3.3 Software Evaluation from the Web Developers 54

5 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS 58

5.1 Summary 58

5.2 Conclusions 59

5.3 Recommendations 60

6 REFERENCES 61

APPENDICES 64

LIST OF TABLES

Table	Title	Page
3-1	Module Name……	29
3-2	Evaluation Criteria for End-Users…….	30
3-3	Evaluation Criteria for Developers……	32
3-4	Score descriptive interpretation…….	36
3-5	Implementation plan for the deployment of the system……	36
4-1	The summary of Functional Testing…….	50
4-2	Perception of the end user on the usability of the website……	52
4-3	Perception of the end user on the effectiveness of the website.	53
4-4	Perception of the end user on the efficiency of the website…..	53
4-5	Perception of the end user on the satisfaction of the website…	54
4-6	Perception of the end user on the context coverage of the website ……..	54
4-7	Summary of results of the website evaluation assessed by the end users ……...	55
4-8	Perception of the IT experts on the functionality of the website……...	55
4-9	Perception of the IT experts on the performance of the website……...	56
4-10	Perception of the IT experts on the usability of the website….	56
4-11	Perception of the respondents on security…….	57
4-12	Perception of the IT experts on the maintainability of the website……...	57
4-13	Perception of the IT experts on the portability of the website..	58
4-14	Summary of results of the website evaluation assessed by the IT experts……...	58

LIST OF FIGURES

Figure	Title	Page
1-1	Conceptual Framework of the study……...	9
3-1	Use Case Diagram of the VisiThoughts…….	27
3-6	Software development model…….	28
4-1	Welcome page……	31
4-2	Welcome transition page……	42
4-3	Form input personal details page……	42
4-4	Form satisfactory level page……..	43
4-5	Form comment section page……..	44
4-6	Form appreciation page……..	44
4-7	Admin starting page……...	45
4-8	Admin login page……...	45
4-9	Admin head dashboard page……..	46
4-10	Admin department records for sentiment……...	47
4-11	Admin client feedback copied form……...	48
4-12	Admin option page…….	48
4-13	Admin setting page……	49
4-14	Admin staff management page……...	50

LIST OF APPENDICES

Appendix	Title	Page
A.	Routing Slip……...	67
B.	Request for Adviser and Technical……	68
C.	Request for Outline Defense……..	69
D.	Request for Final Defense……..	70
E.	Certification of English Critic……	71
F.	Other Appendices……..	72
G.	Evaluation for Developers…….	76
H.	Sample Evaluation for Developers…….	79
I.	Evaluation for End Users……...	82
J.	Sample Evaluation for End Users……...	84
K.	User's Manual……	86
L.	Curriculum Vitae……...	96
N.	Source Code……...	99

VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CVSU-CCAT CAMPUS

ELGENE JOHN V. REYES
KARL DOHNEYZEL A. DELACRUZ
JAN PAOLO A. CANO

An undergraduate thesis presented to the faculty of the Department of Computer Studies at the Cavite State University-CCAT Campus, Rosario, Cavite. In partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science with Contribution No. [CS-0000-00-000]. Prepared under the supervision of Dr. Christopher G. Estonilo and Prof. Jeffrid D. Laxamana.

1. INTRODUCTION

1.1 Project Context

In today's competitive educational landscape, providing high-quality services to customers is of utmost importance for institutions to thrive and succeed. Cavite State University CCAT Campus recognizes the significance of customer feedback in improving and sustaining the quality of services offered by its various offices and departments. However, the institution's ability to gather valuable insights from its customers needs efficient feedback collection.

According to Suciptawati et al. (2019), maintaining customer satisfaction and delivering high-quality service are critical elements of business success. The growth and success of a company heavily rely on its ability to meet customer needs effectively. It is widely acknowledged that providing excellent service quality leads to customer satisfaction, which, in turn, enhances customer retention and fosters loyalty. In addition,

according to Shah et al. (2022), collecting and analyzing customer feedback through online channels has become vital for businesses in retaining existing customers and attracting new ones. Businesses can assess and enhance their offerings by utilizing customer feedback to stay competitive. This feedback mechanism provides valuable insights into customers' preferences and needs and encourages them to share their opinions on products or services. Businesses can identify dissatisfied customers and address their concerns by implementing feedback systems.

To address this issue, the researchers propose developing a responsive web application that will cater to all offices and units of the Cavite State University CCAT Campus. This web application aims to provide an accessible and user-friendly platform for customers and visitors to share their feedback, comments, evaluations, suggestions, and even complaints regarding the services offered by the university and its departments and offices.

The main objective of this research project is to create a system that will enable the collection and analysis of customer feedback across different offices and units of the university. By implementing sentiment analysis techniques, the system will be capable of determining the percentage of negative and positive feedback received by each office or unit. This analysis will provide valuable insights into the areas that require improvement and those performing well.

Additionally, customers will be directed to the proposed web application using QR codes. This approach aims to streamline feedback, ensuring customers can quickly provide input.

In terms of methodology, the research will involve developing and implementing a responsive web application compatible with various devices and platforms. The system will be developed using a Lexicon-based approach. According to Gupta et al. (2020), the lexicon-based approach is used in semantic analysis. This approach determines the sentiment orientation of an entire document or a set of sentences by examining the semantic orientation of the lexicons used. The semantic orientation can be categorized as positive, negative, or neutral. By leveraging the semantic orientation of the lexicons, this technique enables the analysis of sentiment in text at a broader level. The researchers will also conduct user testing and evaluation to ensure the effectiveness and usability of the web application.

While this research project holds significant potential for enhancing the quality of services at the Cavite State University CCAT Campus, anticipated challenges and limitations need to be acknowledged. These may include technical constraints, such as ensuring the security and reliability of the web application, as well as the need for proper training and support for staff members handling the feedback and response process.

In summary, this research project aims to develop a responsive web application that will enable customers and visitors of the Cavite State University CCAT Campus to provide feedback, comments, evaluations, and complaints regarding the services offered by various offices and units. By implementing sentiment analysis, the system will facilitate data-driven decision-making processes and contribute to the continuous improvement of service quality. However, it is crucial to address potential challenges and limitations to ensure the proposed solution's successful implementation and long-term sustainability.

1.2 Purpose and Description

This research project aimed to develop a responsive web application that enables customers and visitors of Cavite State University CCAT Campus to provide feedback, evaluations, and complaints regarding the services offered by various offices and units. The system aims to analyze the feedback received and generate valuable insights for decision-making processes by implementing sentiment analysis techniques.

The primary beneficiaries of this research project were Cavite State University CCAT Campus and its stakeholders, including students, faculty, staff, and administration. By collecting customer feedback systematically and efficiently, the university can identify areas for improvement and take appropriate actions to enhance the quality of services offered across different offices and units. This contributed to a better overall experience for the university community.

The development of a responsive web application for collecting customer feedback had significant potential for positively impacting education and service industries. By implementing sentiment analysis techniques, the system can provide valuable insights into the strengths and weaknesses of various offices and units within the university. This data-driven approach to decision-making can lead to more targeted improvements, optimized resource allocation, and enhanced customer satisfaction.

The significance of this research project extends beyond the Cavite State University CCAT Campus. Educational institutions such as the campus satellites and service-oriented organizations can also benefit from the proposed web application. The web application can serve as a model for similar initiatives, promoting a culture of continuous improvement and customer-centricity. Additionally, the research project can

contribute to advancing sentiment analysis techniques by showcasing their practical application in a real-world setting.

Overall, the successful implementation of this research project can have a transformative impact on the quality of services provided by Cavite State University CCAT Campus and serve as a benchmark for other institutions aiming to enhance customer satisfaction and service excellence.

1.3 Objectives of the Study

The general objective of this study was to develop a responsive web application, VisiThoughts, for collecting customer feedback and conducting sentiment analysis in Cavite State University CCAT Campus.

Specifically, the study aimed to:

1. identify the key requirements and functionalities of the web application through:
    a) conducting a thorough analysis of the feedback collection method and analysis needs of various offices and units within Cavite State University CCAT Campus;
    b) identifying the essential features and functionalities that should be incorporated into the VisiThoughts web application to meet the identified needs; and
    c) defining the user interface design and user experience guidelines to ensure a user-friendly and accessible platform.
2. develop the VisiThoughts web application using Lexicon-based approach.
3. evaluate the application in order to determine if it complies with the modified ISO-IEC 25010 standards.

4. implement the system in CvSU-CCAT Campus.

1.4 Scope and Limitations

The scope of the study encompassed the development and implementation of a responsive web application called 'VisitThoughts' designed explicitly for the Cavite State University CCAT Campus. The primary objective was to create a user-friendly platform where customers and visitors can provide feedback, comments, evaluations, and complaints about the services offered by different offices and units of the university. The web application incorporated sentiment analysis techniques to enhance decision-making processes to analyze the feedback received and generate valuable insights. The development process involved creating a web application that is accessible and compatible with various devices and platforms. The application featured feedback submission forms.

Additionally, integrating QR codes simplified the feedback process, allowing customers to access the web application conveniently. To ensure the effectiveness and usability of the VisitThoughts web application, user testing and evaluation were conducted, incorporating user feedback to make necessary improvements. The study aims to provide a comprehensive solution for collecting customer feedback and leveraging sentiment analysis to enhance service quality at Cavite State University CCAT Campus.

The study acknowledges several limitations that were considered during the development and implementation of the VisitThoughts web application:

The findings and insights generated through sentiment analysis are specific to Cavite State University CCAT Campus. While the system may serve as a model for similar initiatives, the applicability of the results to other educational institutions or service-oriented organizations may vary.

Next, encouraging active user participation and ensuring a sufficient volume of feedback may present a challenge. Efforts should be made to engage customers and visitors, promoting their willingness to provide feedback and contribute to improving services.

Another limitation of this study was that the VisitThoughts web application relies on internet connectivity for users to access and submit feedback. This means that individuals with internet access or in areas with reliable internet connectivity may need help providing feedback through the application. However, departments or units can ask for the customer's email for this intervention. Then, the department or units send the link, which can direct the customers or visitors to the feedback form.

Additionally, the study assumes that the feedback submitted through VisitThoughts is written in English. They may also not effectively capture a context-specific sentiment or sarcasm since they use a lexicon-based approach. Suppose a significant portion of the user base prefers or is more comfortable providing feedback in a different language. In that case, the application may need to capture their input or accurately analyze their sentiments effectively.

1.5 Concept of the Study

The conceptual framework outlines the key components and steps to achieve the study's objectives, as shown in *Figure 1*. The conceptual framework was divided into four (4) phases: (1) Data gathering, (2) Development, (3) Evaluation, and (4) Implementation. The first phase of the conceptual framework involves conducting a thorough analysis of the feedback collection and analysis needs of various offices and units within Cavite State University CCAT Campus. The second phase, which was the development stage, focuses on creating the VisiThoughts web application using the Python programming language, HTML, CSS, Bootstrap, MySQL, and JS using a Lexicon-based approach following the RAD (Rapid Application Development) methodology. The third stage was the evaluation stage; the VisiThoughts application was assessed to determine its compliance with the modified ISO-IEC 25010 standards. The final stage of the framework was implementing the VisiThoughts system in the Cavite State University CCAT Campus.

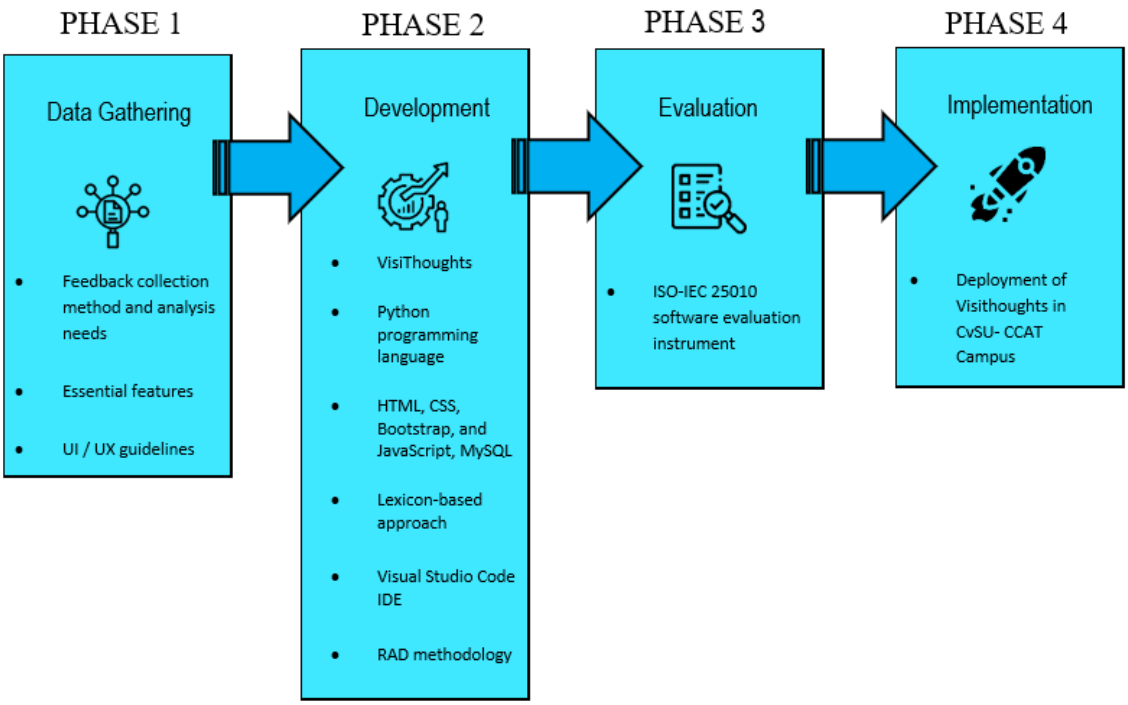


Figure 1-1 Conceptual framework of the study

1.6 Definition of Terms

To understand the study, the below terms are operationally defined:

Sentiments Analysis. Refers to the process of analyzing and determining the sentiment or emotional tone expressed in the feedback. By leveraging sentiment analysis techniques, the project aims to categorize the feedback as positive, negative, or neutral.

Web application. A web application refers to a software application that is accessed and utilized through a web browser or internet connection. The web application will be responsive, compatible with different devices and platforms, ensuring accessibility and user-friendliness.

Feedback. Feedback refers to the opinions, comments, evaluations, and complaints shared by customers and visitors regarding the services provided by Cavite State

University CCAT Campus. By obtaining feedback, the university can identify areas for improvement, address concerns or issues raised by customers, and enhance the quality of services offered. The feedback received through the web application contributed to data-driven decision-making processes and continuous service improvement efforts.

2 REVIEW OF RELATED LITERATURE

This chapter presented the related literatures, studies and systems that the researcher considered in strengthening the claim and importance of the present study.

2.1 Importance of Feedback

According to Morgan et al. (2006), customer feedback is crucial to business performance and improvement. Managers commonly use customer feedback data to set goals and monitor performance on customer satisfaction and loyalty scores. Different customer feedback metrics, such as average customer satisfaction scores and the number of "net promoters," have been advocated and examined to determine their value in predicting future business performance (Morgan et al., 2006). Collecting and analyzing customer feedback is essential because it allows organizations to continuously learn and adapt their offerings to customer preferences (Ordenes et al., 2014). With customers' increasing use of multiple communication channels to provide feedback, organizations face the challenge of developing efficient and effective processes to collect and analyze all the information (Ordenes et al., 2014). Feedback helps promote professional and personal growth, adding value to customer satisfaction (Kerdpitak, 2021). Listening to customers, analyzing the problem, and identifying better solutions are part of the feedback process (Kerdpitak, 2021). However, customers may withhold feedback due to underestimating the other side's desire for feedback. Organizations must implement well-designed complaint-handling procedures and avoid pressuring customers to evaluate their services (Obradović et al., 2022).

2.2 Sentiment Analysis in Interpreting Feedback

Sentiment analysis is crucial in interpreting feedback across various domains, including investment behavior, product quality assessment, marketing analysis, and customer satisfaction. Businesses and researchers can gain valuable insights into customer opinions, attitudes, and needs by analyzing the sentiment expressed in user reviews and feedback. For product managers, sentiment analysis can assist in interpreting customer feedback and improving the quality of products. By analyzing user reviews, sentiment analysis models can classify the sentiment expressed in the reviews, providing valuable information about customer satisfaction and areas for improvement (Khalid et al., 2020). In marketing analysis, sentiment analysis is used to understand customer perspectives and opinions. It helps assess the sentiment toward individuals, events, products, and topics, providing insights into customer attitudes and preferences (Kayıkçı, 2022).

Similarly, sentiment analysis is also valuable in the hospitality and tourism industry, where online review platforms serve as valuable sources of customer feedback. By applying sentiment analysis techniques, marketers can obtain firsthand information about customers' feedback and needs (Rita et al., 2022). Furthermore, education has applied sentiment analysis to analyze student feedback and satisfaction. Using sentiment-based approaches, aggregated sentiment scores can be used as inputs to fuzzy logic modules for analyzing student feedback and improving satisfaction (Asghar et al., 2019). In social media, sentiment analysis is crucial for understanding user evaluation and content generation. By decomposing user reviews into different dimensions and incorporating them into econometric models, sentiment analysis helps measure service quality, shape user evaluation, and drive content generation (Duan et al., 2015). However, it is essential

to note that while sentiment analysis models can achieve high accuracy, many of them lack interpretability. This poses a challenge in terms of trust and explainability of the system. To address this issue, implementing explainable artificial intelligence (XAI) techniques can enhance the interpretability of sentiment analysis models (Adak et al., 2022).

2.3 Lexicon-based Approach for Sentiment Analysis

The lexicon-based approach for sentiment analysis is a popular method that uses sentiment lexicons to determine the sentiment polarity of text. Sentiment lexicons are dictionaries or lists of words associated with sentiment labels or scores, indicating whether the words have a positive or negative sentiment. This approach relies on the assumption that the sentiment of a text can be inferred from the sentiment of its constituent words (Htait et al., 2021).

One study by Yang et al. (2020) applied the lexicon-based approach to sentiment analysis of e-commerce product reviews in Chinese. They used a sentiment lexicon and deep learning techniques to analyze the sentiment expressed in the reviews (Yang et al., 2020). Another study by Chundi et al. (2023) proposed a lexicon-based approach called NBLex for sentiment analysis of Kannada-English code-switch text. This study used the lexicon-based approach to perform sentiment analysis in a multilingual context (Chundi et al., 2023).

The lexicon-based approach can be compared with other methods, such as machine learning-based approaches. Dhaoui et al. compared the sentiment analysis method based on lexicon and machine learning using the LIWC2015 lexicon and RTextTools machine learning package (Yang et al., 2020). They found that the lexicon-based approach achieved comparable performance to the machine learning-based approach.

In addition to sentiment lexicons, other lexicons can enhance sentiment analysis. For example, Navarrete et al. (2021) proposed a novel approach to creating a labeling lexicon for improving emotion analysis in the text. They used Plutchik's eight emotions as labels in the lexicon to capture a broader range of emotional expressions (Navarrete et al., 2021).

2.4 Effectiveness of Web Application for Collecting Feedback

The effectiveness of web applications for collecting feedback has been studied in various domains. Mofidi et al. (2019) proposed an integrated model that collects occupants' feedback through participatory sensing using smartphones or web-based applications in office productivity and energy cost optimization. The authors defined productivity as "the extent to which activities have provided performance regarding system goals." This suggests that web applications can effectively gather feedback from occupants to optimize office productivity and energy costs.

Selvam et al. (2022) highlighted the critical issues in feedback collection in the context of web-based learning applications. They emphasized the importance of test components that evaluate the user, provide information about user performance, encourage the user, and focus on user interaction with the computer. This indicates that web-based applications can effectively collect feedback by incorporating these essential functions.

In the context of student feedback collection, Wambsganss et al. (2022) discussed the limitations of static and old-fashioned web surveys. They emphasized the need for interactive and engaging designs to meet the expectations of young students. This suggests that web applications with interactive features can effectively collect feedback from students and improve their satisfaction rates.

Nguyen et al. (2019) conducted a systematic review of eHealth interventions addressing HIV/STI prevention. They found that web-based interventions, smartphone applications, SMS/Text messages, and live-chat/chat room approaches allow real-time monitoring, tailored feedback, and interactive person-to-person conversations. These features are critical for building rapport between interventionists and participants, indicating the effectiveness of web applications in collecting feedback for HIV/STI prevention.

Lobchuk et al. (2023) conducted usability testing of a web-based empathy training portal. They collected feedback through various application features, such as self-awareness exercises, instruction, feedback, and self-evaluation with video feedback. This study demonstrates the effectiveness of web applications in collecting feedback for training purposes.

2.5 Python in sentiments analysis

Python is an essential programming language for creating sentiment analysis web applications. Sentiment analysis is a natural language processing (NLP) technique that involves extracting opinions, emotions, and attitudes expressed in text (Devi, 2023). With its extensive libraries and frameworks, Python has become one of the most popular programming languages for NLP tasks, including sentiment analysis (Devi, 2023).

Python's versatility and ease of use make it suitable for implementing sentiment analysis algorithms and machine learning models (Hassan et al., 2020). Researchers have developed sentiment analysis algorithms in Python using data mining and machine learning techniques. Python libraries like MonkeyLearn and SVM algorithms have been used for

sentiment analysis in various domains, including the hotel industry and environmental management (Saura et al., 2019).

Python's data analysis capabilities are also valuable for sentiment analysis in the financial sector. In a study comparing traditional and digital methodologies for brand communication analysis in the financial sector, sentiment analysis using Python allowed for more accessible and faster data analysis than traditional methods (Saura et al., 2019).

2.6 Bootstrap, HTML, CSS, and Java Script for creating web application

Bootstrap, HTML, CSS, and JavaScript are essential for creating web applications. Bootstrap is a popular framework that combines CSS, HTML, and JavaScript to create responsive webpages (Rahayuda et al., 2021). It provides templates and components that developers can use to design web pages that adapt to different screen sizes and devices (Wang et al., 2023). Using Bootstrap, developers can easily create a consistent and visually appealing user interface for their web applications.

HTML (Hypertext et al.) is the standard markup language for creating web pages. It provides the structure and content of a webpage (Bajaj et al., 2014). CSS (Cascading Style Sheets) is used to style and format the HTML elements, improving the presentation and visual appeal of the webpage (Bhargav N & Jayanthi N, 2021). Conversely, JavaScript is a programming language that adds interactivity and dynamic behavior to web pages (Bajaj et al., 2014). It allows developers to manipulate HTML elements, handle events, and perform calculations and data processing on the client side.

Together, these technologies enable the creation of modern web applications. HTML provides the structure, CSS enhances the presentation, and JavaScript adds interactivity and functionality to the web pages (Bajaj et al., 2014). As a framework,

Bootstrap simplifies the process of designing responsive web layouts and ensures a consistent user experience across different devices (Santoso, 2019).

2.7 ISO EIC 25010

ISO/IEC 25010 is a widely used evaluation tool for assessing the quality of web applications. It is a quality-in-use model with five characteristics; each further divided into sub-characteristics. These characteristics contribute to the overall interaction outcome when a product is used in a specific sense of use. The ISO/IEC 25010 model encompasses the entire human-machine system, including the computer systems and the software products in use (Abana et al., 2022).

The ISO/IEC 25010 model is comprehensive and covers various aspects of software quality. It includes functional suitability, reliability, performance efficiency, usability, compatibility, security, maintainability, and portability (Malaya et al., 2022). These characteristics provide a holistic view of a web application's quality, considering its functional and non-functional aspects, such as performance, security, and maintainability.

ISO/IEC 25010, as an evaluation tool, allows for a standardized and systematic assessment of web applications. It provides a framework for measuring and evaluating the quality of a web application based on international standards (Malaya et al., 2022). Organizations can use ISO/IEC 25010 to ensure that their web applications meet the desired quality standards and deliver a satisfactory user experience.

Several studies have utilized ISO/IEC 25010 for evaluating web applications. For example, in evaluating a rice seed analyzer application, ISO/IEC 25010 was used to assess the effectiveness of the application based on its quality characteristics (Abana et al., 2022).

Similarly, in evaluating an information management system, ISO/IEC 25010 was used to calculate the quality aspect of the system (Malaya et al., 2022).

The ISO/IEC 25010 model has also been extended to incorporate new characteristics and concepts, such as information quality, learnability, actual usability, and user experience (Lew et al., 2010). This extension allows for a more flexible and integrated approach to evaluating web applications, considering their technical aspects, usability, and user experience.

2.8 SQLite as web server for web applications

SQLite is a capable web application server when used in conjunction with Python and the Django framework. Several studies have demonstrated the successful implementation of web applications using SQLite as the database tool integrated with Django (Guebila et al., 2021). For example, a literature web application was developed using Python, Django, and SQLite, and a recommendation system was implemented to provide accurate suggestions (Kozuyla et al., 2021). Another study developed a web-based leave management system using web technologies such as HTML, CSS, Python, and Django, with the database designed using SQLite (Mayowa et al., 2022). A web application for storing information about students' theses was also developed using Python, Django, and SQLite (Karpei et al., 2021). These examples highlight the versatility and effectiveness of SQLite in web application development when combined with Python and the Django framework.

2.9 Importance and benefits of using Django in creating web application with sentiment analysis

Django is a popular web development framework that offers several benefits for creating web applications with sentiment analysis. One of the key advantages of using Django is its ability to handle complex database-driven sites (Alkot, 2022). This is particularly useful for web applications that involve sentiment analysis, as they often require storing and retrieving large amounts of data related to user profiles, sentiment scores, and text analysis results.

In creating a web application for sentiment analysis, Django's Model-View-Template (MVT) architecture is highly beneficial (Dhanalakshmi et al., 2023). The MVT architecture separates the application's data models, user interface, and business logic, making managing and maintaining the codebase easier. This separation allows developers to focus on implementing the sentiment analysis algorithms and user interface components without worrying about the underlying database operations and request handling.

Additionally, Django provides a robust and secure authentication system, which is crucial for web applications that involve user-profiles and sensitive data (Dhanalakshmi et al., 2023). This feature ensures only authorized users can access and modify the sentiment analysis results and other related information. It also allows user roles and permissions to be implemented, enabling different access levels for job seekers, recruiters, and administrators (P et al., 2023).

Furthermore, Django offers a wide range of built-in tools and libraries that can facilitate sentiment analysis tasks. For example, Django's form-handling capabilities can create user-friendly interfaces for inputting and submitting text for sentiment analysis

(Dhanalakshmi et al., 2023). The framework also supports integration with external libraries and APIs, which can be leveraged to incorporate state-of-the-art sentiment analysis algorithms and models into the web application (Atzeni et al., 2022).

Regarding scalability and performance, Django provides features such as caching, database optimization, and load balancing, which are essential for handling large volumes of data and user requests in real time (Dhanalakshmi et al., 2023). These features ensure that the sentiment analysis web application can handle high traffic and deliver fast response times, even when processing complex sentiment analysis tasks.

2.10 Automating Large Scale Healthcare Service Feedback Analysis: Using Natural Language Processing and Topic Modelling (Preprint)

The related system focuses on developing a data analysis and interactive visualization pipeline for patient feedback in England's National Health Service (NHS). The system uses natural language processing and machine learning techniques to analyze free-text clinical service reviews from patients. It incorporates sentiment analysis and topic modeling to derive meaningful insights from the feedback data. The system also includes an interactive web application that presents the analyzed data on a geographic representation, making it easily accessible to the general public.

2.11 Opinion mining technique for developing student feedback analysis system using lexicon-based approach (OMFeedback)

The related system focuses on online assessment systems used for evaluating the performance of lecturers in Institutions of Higher Learning (IHLs). It addresses the issue of paying attention to the qualitative part of the assessment, which includes students' opinions, due to the challenges of manually analyzing unstructured texts. The feedback

system utilizes an opinion-mining technique supported by a lexicon-based approach to analyze students' feedback automatically. It uses the Vader Lexicon, a database of English sentiment words, to determine the polarity of words and quantify the sentiment tendency of students' feedback. By analyzing these sentiment words, including intensifier words and additional features like capitalization and emoji characters, the system generates results for the lecturer teaching assessment, indicating the level of positive, negative, or neutral opinions. The system aims to provide valuable information to IHLs for improving lecturers' teaching skills and course materials.

2.12 Web-Based Student Opinion Mining System Using Sentiment Analysis

The related system described focuses on developing a student opinion mining system to streamline the process of collecting and analyzing feedback from students in educational institutions. The system aims to provide a faster and easier feedback-gathering method while ensuring confidentiality. It utilizes tools such as Python, Scikit-learn, Textblob, Pandas, and SQLite for development and Django for hosting the application on a local host server. The system employs sentiment analysis techniques, using a dataset from online feedback forms distributed to students. Feature selection using the chi-square method and sentiment classification using the support vector machine algorithm are implemented. The application aims to assist university administrators and lecturers in identifying the strengths and weaknesses of lecturers based on the textual evaluations provided by students.

2.13 Sentiment Analysis for University Students’ Feedback

The system focuses on developing a system to automatically categorize students' feedback in higher education institutions in Vietnam. Automating the feedback analysis

process, the system aims to save time, human resources, and costs. The study begins by creating a dataset of university students' feedback over two years and organizing them into three classes: Positive, Negative, and Neutral. A Vietnamese sentiment dataset with 5000 classified sentences is built for training and testing. Three classifiers, Naïve Bayes, Maximum Entropy, and Support Vector Machine, are utilized on the annotated data. The results indicate that the Maximum Entropy algorithm performs better than Naïve Bayes and Support Vector Machine, achieving an accuracy score of 91.36%. Based on these findings, the system is implemented to detect students' opinions, allowing for adjustments and improvements in lectures, facilities, and curriculum to enhance the quality of education over time.

2.14 Sentiresearch: Lexicon- based web application for indonesian sentiment analysis

The related system focuses on developing a system to automatically categorize students' feedback in higher education institutions in Vietnam. Automating the feedback analysis process, the system aims to save time, human resources, and costs. The study begins by creating a dataset of university students' feedback over two years and organizing them into three classes: Positive, Negative, and Neutral. A Vietnamese sentiment dataset with 5000 classified sentences is built for training and testing. Three classifiers, Naïve Bayes, Maximum Entropy, and Support Vector Machine, are utilized on the annotated data. The results indicate that the Maximum Entropy algorithm performs better than Naïve Bayes and Support Vector Machine, achieving an accuracy score of 91.36%. Based on these findings, the system is implemented to detect students' opinions, allowing for

adjustments and improvements in lectures, facilities, and curriculum to enhance the quality of education over time.

Compared to our study, the related system focuses on automating the categorization of students' feedback, specifically in higher education institutions in Vietnam. It aims to improve efficiency by automatically analyzing feedback and providing insights for quality improvement. On the other hand, our study revolves around developing the responsive web application "VisiThoughts" for collecting customer feedback in the context of the Cavite State University CCAT Campus. Our study aims to enhance the quality of services offered by various offices and units within the university by implementing sentiment analysis techniques. Both systems involve feedback analysis and aim to improve the quality of education or services; the research also used a lexicon-based approach, python, and Django in developing their system.

2.15 Synthesis

The literature and studies referenced in the research were intended to provide backing for the development of the study. The overarching theme across various contexts is the significance of feedback collection and sentiment analysis in different domains. Customer feedback was emphasized as a crucial asset for businesses, enabling them to predict performance, adapt offerings, and enhance overall customer satisfaction. Sentiment analysis is pivotal for interpreting feedback and providing valuable insights into opinions and attitudes.

The discussion extended to the methodologies employed in sentiment analysis, such as the lexicon-based approach, highlighting its versatility, applicability across languages and domains, and potential for comparative evaluations with other methods. Additionally,

the need for explainability in sentiment analysis models was addressed, emphasizing the implementation of eXplainable Artificial Intelligence (XAI) techniques to enhance trust.

These technologies' application was exemplified in creating a web application named ‘VisiThoughts’ for Cavite State University-CCAT Campus. This application aimed to collect customer feedback, employ sentiment analysis techniques and enhance the quality of services within the university. This acknowledges the uniqueness of this study compared to related systems, emphasizing the specific focus on an educational institution, distinct functionalities, and the potential impact on service quality improvement.

3 MATERIALS AND METHODOLOGY

This section discussed the materials, hardware and software, data, and the methodology that will be using by the researcher.

3.1 Materials

3.1.1 Software

Front-End Technologies:

- HTML: Markup language for creating the structure of web pages.
- CSS: Style sheet language for designing the visual presentation of web pages.
- JavaScript: Programming language for implementing interactivity and client-side functionality.
- Bootstrap: Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites.

Back-End Technologies:

- Server-Side Language: Python for implementing the back-end logic.
- Web Framework: Django (Python), to help structure and streamline your back-end code.
- Database: MySQL provides a robust and scalable solution for storing and synchronizing data in real-time. With its robust support for standard SQL syntax and efficient handling of concurrent transactions, MySQL offers a powerful and reliable database management system.
- Sentiment Analysis Library: A lexicon-based sentiment analysis library or tool will be required to analyze the sentiment of user feedback.

- NLTK (Natural Language Toolkit): A Python library that provides various tools and resources for natural language processing, including sentiment analysis.

3.1.2 Hardware

In this section, described the hardware that used for the development and testing of the responsive web application. The hardware requirements include:

Development Computer: A computer system capable of running IDE and supporting the necessary software development tools.

Android/Computer Devices: Devices such as smartphones and Personal Computer running the Android and windows operating system. These devices used for testing and evaluating the responsive web application. The specifications of said devices were yet to be determined upon development.

3.1.3 Data

In this section, the researchers indicated the data source, data type, year of acquisition, and other pertinent details. The data required for developing the responsive web application was gathered from the Cavite State University-CCAT Campus. Data such as the name and services offered of the different departments and offices present on the Campus, the basic information of the staff like their name and position in the departments, the different services that the departments offered, and gathered from Cavite State University – CCAT Campus through document request/documentation or interview.

3.2 Methodology

The major phases of the conceptual framework are discussed in the methodology.

3.2.1 Data Analysis

The study involved processing and analyzing the gathered data to extract meaningful insights and make informed decisions for developing VisiThougths. The data analysis process primarily focuses on analyzing the gathered data in the CvSU-CCAT Campus to determine the departments, units, offices, staff, and services needed for web application development.

3.2.2 Development of the Application

The following discussed the design of the software, system, application, or processes and how the application developed using the software development model RAD (Rapid Application Development). It also includes the functional testing.

a) Design of the Software, System, Application, Product or Processes

Figure 3-1 shows the use case diagram of the system. There are 3 actors such as the User or Visitor, the Chairperson or Unit Head, and the Admin.

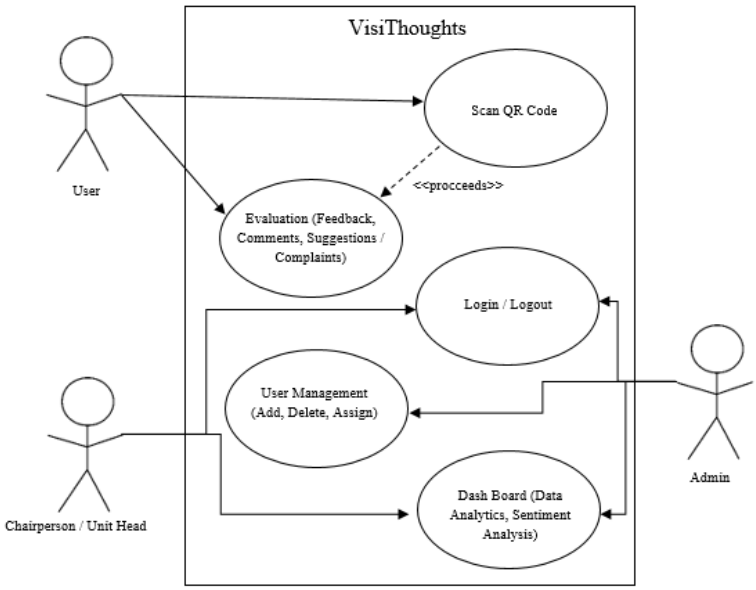


Figure 3-1 Use Case Diagram of the VisiThoughts

b) Software Development Model

The application was developed using the Rapid Application Development Model (RAD) shown in Figure 3-2. Also, to test the application, the researchers used functional testing. The RAD (Rapid Application Development) model was an iterative and incremental software development approach focusing on quickly building and refining prototypes based on testing, assessment, and feedback. In developing the web application system, here is how the RAD model was applied in developing the VisiThoughts. In requirement planning, identify specific requirements for the web application, focusing on feedback processing and sentiment analysis using a lexicon-based approach. Clearly define the features and functionalities related to sentiment analysis.

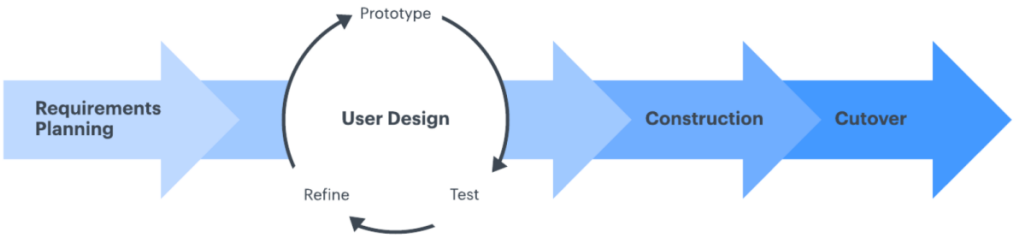


Figure 3-2 Software development model

In the user design cycle, develop the web application prototype that includes the core feature of sentiment analysis, and the system receives input text for analysis and displays sentiment results. Testing: the researchers test the prototype, specifically evaluating the accuracy of sentiment analysis results. Identify any issues or improvements needed in the initial implementation. Iterate: the researchers, based on testing results, iterate on the sentiment analysis algorithm, refine the prototype, and ensure that the analysis accurately determines whether the text is positive, neutral, or negative.

In rapid construction the researchers begin rapid construction of the web application, incorporating refined sentiment analysis. Develop additional features and components necessary for the complete web application. Maintain the ability to revisit the user design phase if substantial feedback or changes are required through evaluation. Sample users' and web developers' feedback triggers design and sentiment analysis adjustments. Use feedback to drive further improvements in subsequent iterations.

In cutover, the researchers prepare for the web application's final deployment and launch, ensuring all features and functionalities, including sentiment analysis, are thoroughly tested and ready for production. In summary, the RAD model allows for a dynamic and iterative web application development approach.

c) Functional Testing

Table 3-1 Module Name

Module			**Account**		
Test Case	**Action**	**Expected System Response**	**Number of Execution**	**Results**	**Status**

The system was tested using functional testing as shown in Table 3-1.

3.2.3 Evaluation of the Application

Table 3-2 Evaluation Criteria for End-Users

Indicator	5	4	3	2	1
A. Usability					
1. Appropriateness recognizability - degree to which users recognize the appropriateness of the website based on their needs					
2. Learnability - degree to which the website provides self-explanatory user-interface; entry fields contain default/placeholder texts; and error messages provided are understandable					
3. Operability - degree to which the website is easy to operate, control, and can be used appropriately					
4. User error protection - the website is able to perform user entry validation and allows the user to recover from errors					
5. User interface aesthetics - degree to which the user interface enables pleasing and satisfying interaction with the user					
B. Effectiveness					
1. The website allows the users to achieve specified goals when used in specified contexts					

Table 3-2 contd.

C. Efficiency 1. The degree to which resources are consumed with accuracy and completeness of helping users achieve the goals	
D. Satisfaction 1. Usefulness - degree to which the user is satisfied on perceived achievement of pragmatic goals including the results and consequences of using the website 2. Trust - degree to which a user or other stakeholders are confident that the website will behave as it is intended 3. Pleasure - degree to which a user gains pleasure upon fulfillment of personal needs 4. Comfort - degree to which the user is satisfied with physical comfort	
E. Context Coverage 1. Content completeness - degree to which the website can be used effectively, efficiently, and satisfactorily in all specified contexts of use 2. Flexibility - degree to which the website can be used effectively,	

Table 3-2 contd.

efficiently, and satisfactorily in contexts beyond those initially specified in the requirements					

The system was evaluated in order to determine if it complies with the modified ISO-IEC 25010 standards. Table 3-2 are the evaluation criteria for end-users, and Table 3-3 are the evaluation criteria for the web developers or IT experts.

Table 3-3 Evaluation Criteria for Developers

Indicator	5	4	3	2	1
A. Functional Suitability					
1. Functional completeness - degree to which the set of functions covers all the specified tasks and user objectives					
2. Functional correctness - degree to which the functions provide the correct results with precision					
3. Functional appropriateness - degree to which the functions facilitate the accomplishment of specified tasks and objectives					
B. Performance Efficiency					
1. Time behavior - degree to which the responses, processing times, and throughput rates of the website meet the requirements when performing its functions					
2. Resource utilization - degree to which the amounts and the types of resources used by the website meet the requirements when performing its functions					

Table 3-3 contd.

3. Capacity - degree to which the maximum limitations of the website or its components meet the requirements

C. Usability

1. Appropriateness recognizability - degree to which users recognize the appropriateness of the website based on their needs

2. Learnability - degree to which the website enables the user to learn how to use the software effectively

3. Operability - degree to which the website is easy to operate, control, and be used appropriately

4. User error protection - degree to which the website protects the user from making errors

5. User interface aesthetics - degree to which the user interface enables pleasing and satisfying interaction with the user

6. Accessibility - degree to which the website can be used in a specific context with the widest range of characteristics and capabilities to achieve a specific goal

D. Security

Table 3-3 contd.

1. Confidentiality - degree to which the website ensures that the data are only accessible by those who have authorized access

2. Integrity - degree to which the website prevents unauthorized access to and or modifications of computer programs or data

3. Non-repudiation - degree to which actions or events can be proven to have taken place to avoid repudiation later on

4. Accountability - degree to which the actions of an entity can be uniquely accounted to the entity

5. Authenticity - degree to which the identity of a subject or resource can be proven to be the one being claimed

E. Maintainability

1. Modularity - degree to which the website consists of discrete components so that a change on one component pose minimal impact to other components

2. Reusability - degree to which an asset can be used in more than one system or in building other assets

Table 3-3 contd.

3. Analyzability - degree of effectiveness and efficiency on possibly assessing the impact of an intended change to one or more parts of the website; diagnosing deficiencies or causes of failures; and or identifying parts of the website to be modified	
4. Modifiability - degree to which the website can be effectively and efficiently modified without introducing defects or degrading existing quality	
5. Testability - degree in which test criteria can be efficiently and effectively established and the software or components can be tested to determine whether test criteria were met	
F. Portability	
1. Adaptability - degree to which the website can be effectively and efficiently adapted for different or evolving hardware, software, or other operational or usage environments.	

Table 3-4 Score descriptive interpretation

SCORE	DESCRIPTIVE INTERPRETATION
4.51 – 5.00	Excellent
3.51 – 4.50	Very Satisfactory
2.51 – 3.50	Satisfactory
1.51 – 2.50	Unsatisfactory
1.00-1.50	Requires Improvement

Table 3-4 shows the score descriptive interpretation used in interpreting the results.

3.2.4 Deployment Plan

Table 3-5 Implementation plan for the deployment of the system

Strategy	Activities	Person Involved	Duration
Approval from the Campus Administrator	Letters for the Administrator	Researchers, Administrator	1 Day
System's Installation	Installation of the System and required software	Researchers, Administrator	5 Hours
Information Distribution	Manuals	Researchers	1 Day
End-User Training	Hands on Training and Lectures	Department Chairperson and Unit Heads, Faculty, Staff, and Students, Stakeholders	3 Days

The researchers deployed the responsive web application in all Department and Offices of Cavite State University-CCAT Campus provided with their services, activities, persons involved, and duration as shown in Table 3-5.

4 RESULTS AND DISCUSSION

This chapter offers a comprehensive review of the results obtained during the different stages of the system's development.

4.1 Results of the data analysis and model building

Lexicon-Based Approach: A lexicon-based approach to sentiment analysis involved using a predefined set of words or phrases with associated sentiment scores to determine the sentiment expressed in a piece of text. In this approach, a sentiment lexicon, which was essentially a dictionary of words or expressions along with their corresponding sentiment polarity (positive, negative, or neutral), used to analyze the sentiment of a given document, sentence, or even individual words.

NLTK Vader Library: The Natural Language Toolkit (NLTK) Vader library was widely used tool for sentiment analysis in natural language processing. It adopts a lexicon-based approach, relying on a pre-built sentiment lexicon and a set of rules to determine the sentiment of a given text.

Sentiment Lexicon: The core component of the NLTK Vader library was its sentiment lexicon. This lexicon comprises a vast collection of words, each assigned a sentiment score indicating the degree of positivity, negativity, or neutrality. The lexicon was pre-constructed through the amalgamation of human-labeled data and empirical analysis.

Scoring Mechanism: Words in the lexicon were assigned polarity scores ranging from highly negative to highly positive. The cumulative score of words in a given text determines the overall sentiment.

Intensifiers and Negators: The NLTK Vader lexicon accounts for intensifiers and negators, which modify the sentiment of adjacent words. This helps capture nuances in sentiment, such as the impact of words like "very" or the reversal effect of negations like "not."

Development: The integration of the NLTK Vader library into the Django web application streamlines the sentiment analysis process. By leveraging a lexicon-based approach, the model efficiently analyzes user comments, providing a clear and interpretable sentiment label. The combination of preprocessing, lexicon-based sentiment analysis, and score aggregation allows for a robust understanding of sentiment in user feedback, contributing to enhanced user engagement and improved decision-making within the web application.

1. *Data Preprocessing*: Upon receiving user comments through the Django form, the initial step was to preprocess the text for better analysis. The following preprocessing steps are applied:

a) *Lowercasing*: Convert all text to lowercase to ensure uniformity and avoid discrepancies due to case differences.

b) *Punctuation Removal*: Stripping away punctuation ensures that the sentiment analysis focuses on the content words, reducing noise in the text.

c) *Tokenization*: Breaking down the comments into individual words, or tokens, was a fundamental step for further analysis. Tokenization was essential for understanding the sentiment of each word in the comment.

2. *NLTK Vader Sentiment Analysis*: The NLTK Vader library was then employed to conduct sentiment analysis on the preprocessed text. The library utilizes a pre-

built sentiment lexicon, which consists of words with associated sentiment scores. Each word in the tokenized comment was matched with its sentiment score from the lexicon.

Scoring Mechanism: The sentiment analyzer in NLTK Vader calculates sentiment scores for each comment. These scores include a positive score, a negative score, a neutral score, and a compound score that represents the overall sentiment intensity.

3. *Aggregating Sentiment Scores*: Following sentiment analysis, the next step was to aggregate the sentiment scores and determine an overall sentiment label for the comment. The approach taken involves comparing the negative and positive scores.

Comparing Scores: If the negative score was higher than the positive score, the comment was labeled as having a negative sentiment. Conversely, if the positive score was higher, the sentiment is classified as positive. In cases where the positive and negative scores are equal, the comment was considered neutral.

4. *Returning Sentiment Label*: Based on the aggregated sentiment score, the final step was to return a sentiment label to the Django web application. This label was utilized for various purposes, such as categorizing feedback or generating insights into user sentiments.

Practical Considerations for Applying Sentiment Analysis Results in the Web Application:

1. *User Feedback Sorting and Prioritization*: Utilize sentiment labels to automatically categorize and prioritize user feedback. Positive feedback can be highlighted for visibility, while negative feedback may be flagged for urgent attention.

2. *Feedback Dashboard*: Implement dashboards that visually represents the sentiment gathered through feedback. This allows administrators to quickly view sentiments and feedbacks to produce and use the information.

3. *Product or Service Improvement Insights*: Analyze common themes in negative comments to identify areas for improvement. Use sentiment analysis to inform product or service development strategies and prioritize enhancements that align with user feedback.

Potential Improvements and Modifications:

1. *Handling Sarcasm and Irony*: Develop strategies to better handle sarcasm and irony in user comments. This could involve exploring more advanced sentiment analysis models that consider contextual cues for nuanced expressions.

2. *User Feedback Integration into Decision-Making*: Explore ways to integrate sentiment analysis results directly into decision-making processes. For instance, use sentiment trends to inform business strategy or guide marketing campaigns.

4.2 Results of the application development

Figure 4-1 shows the welcome page of the system, this page shows that the clients will need to proceed to evaluate and use the form.

Figure 4-1 Welcome page

Figure 4-2 shows the transition of the welcome page. This page indicates that clients should proceed with the assessment and utilization of the form.

Figure 4-2 Welcome Transition page

Figure 4-3 shows the users info page. This page delineates the requirement for users to input their personal details, serving the dual purpose of information gathering and establishing their identity as clients.

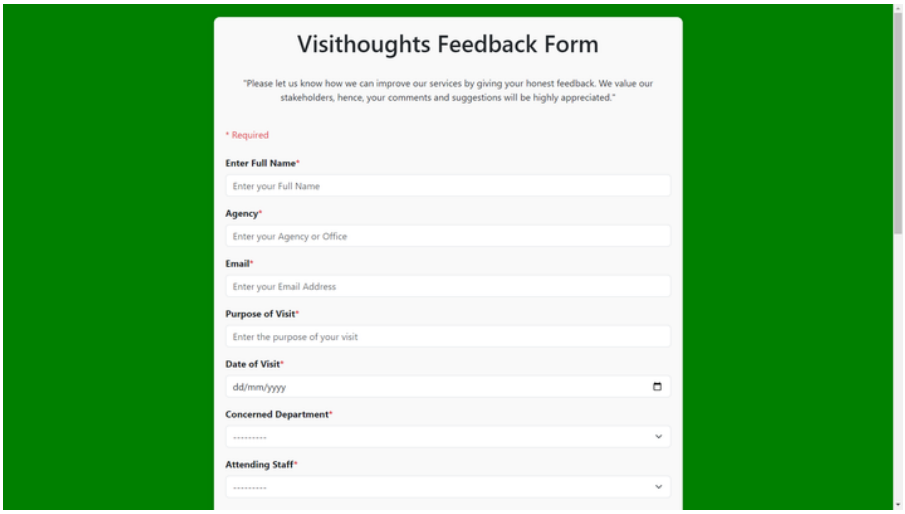


Figure 4-3 Form input personal details page

Figure 4-4 shows the presents users with a slider interface for the evaluation process. This page instructs users to assess the performance of the staff services by simply utilizing a slider mechanism to indicate their level of satisfaction.

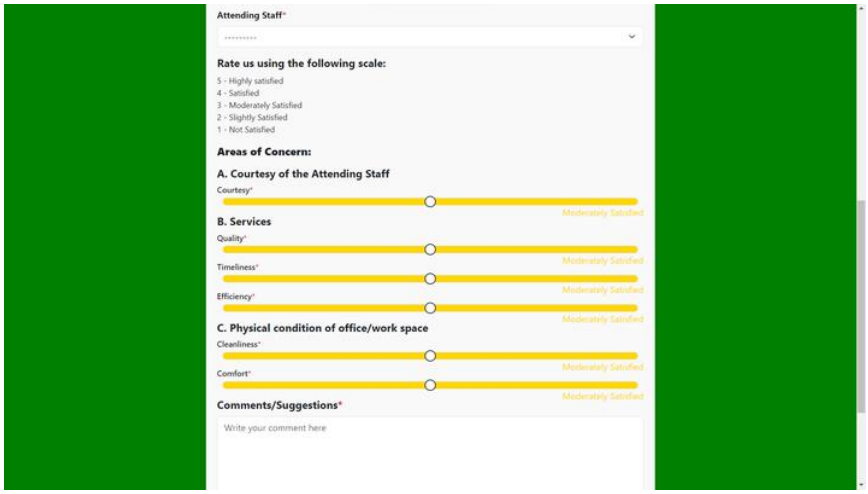


Figure 4-4 Form satisfactory level page

Figure 4-5 shows the comment section satisfactory level of the users for the evaluation process. This page features a comment section within the form where clients are required to provide feedback, expressing either positive or negative sentiments based on their satisfaction.

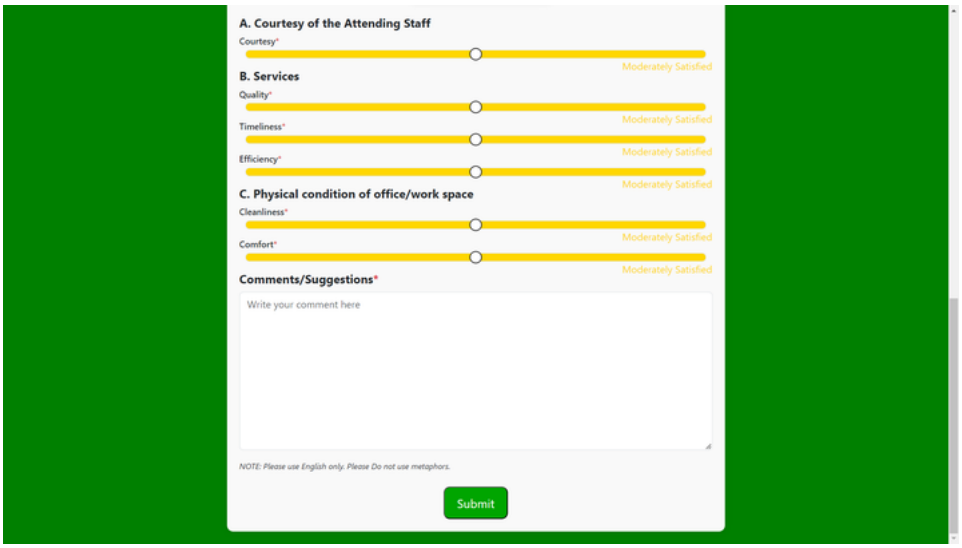


Figure 4-5 Form comment section page

Figure 4-6 shows an appreciation page upon the completion of the client's feedback form. This page expresses gratitude for the client's time and valuable input in providing feedback, emphasizing the significance of their opinions in assessing the service quality of each office.

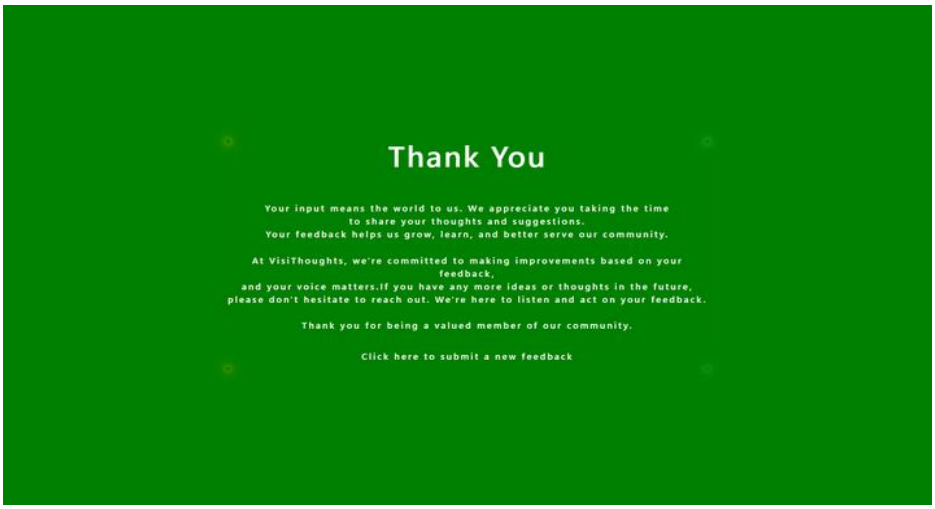


Figure 4-6 Form appreciation page

Figure 4-7 shows the starting page of the admin users. This page indicates that administrators are required to initiate the login process by clicking the start button in order to access their accounts and manage their schedules within the system.

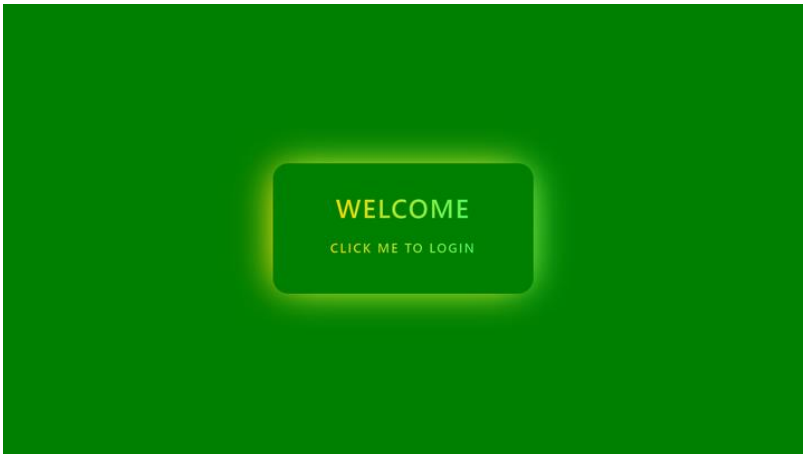


Figure 4-7 Admin starting page

Figure 4-8 shows the admin login page. This page serves as the gateway for administrators to securely log in, providing a streamlined and secure means for them to enter their account information and gain authorized entry to the system's functionalities.

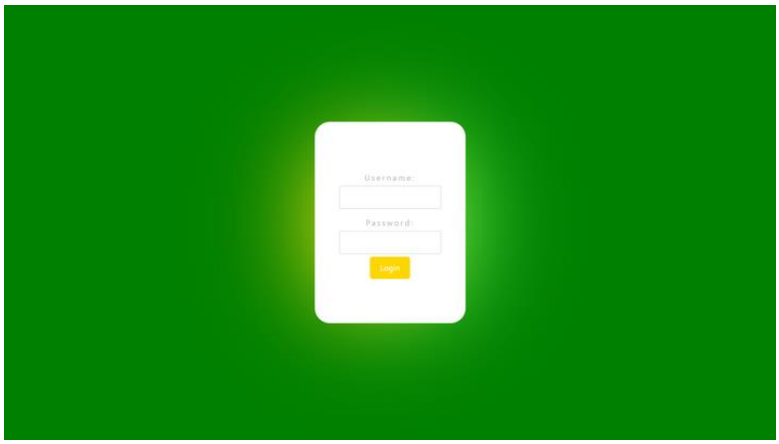


Figure 4-8 Admin login page

Figure 4-9 shows the admin head dashboard page. This page allows administrators to conveniently categorize specific details according to their preferences, particularly with respect to dates, whether for an overarching assessment of the campus or for individual evaluations of each department and staff member. Additionally, the platform provides insights into satisfaction evaluations and sentiment analysis results, offering a comprehensive overview of the data at hand.

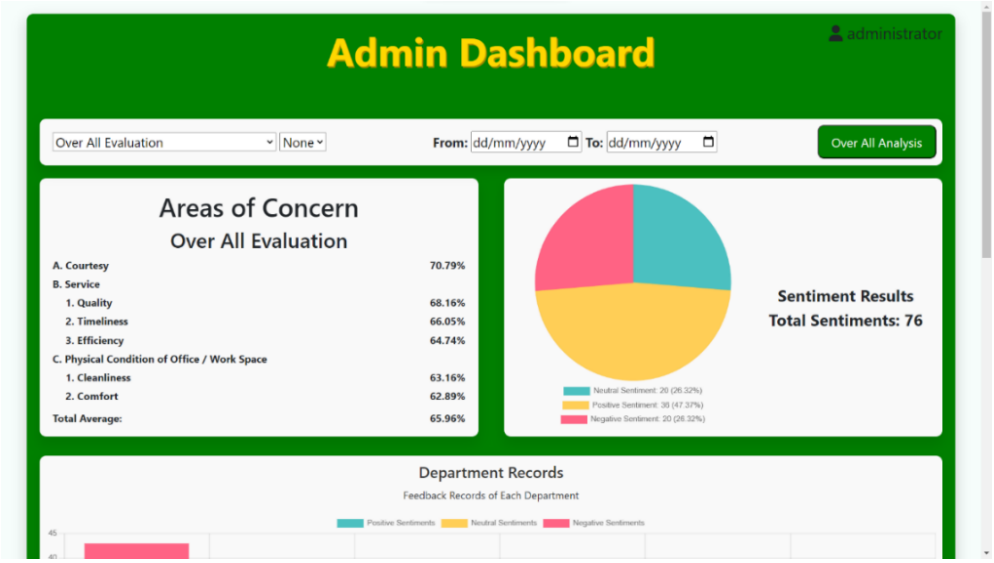


Figure 4-9 Admin head dashboard page

Figure 4-10 shows the Department record page. This page displays the sentiment status of each department within the CvSU-CCAT Campus, offering the capability to assess overall sentiments categorized into positive, negative, and neutral classifications. The presentation is designed to ensure clarity for administrators, employing distinct color coding for each result to mitigate any potential confusion.

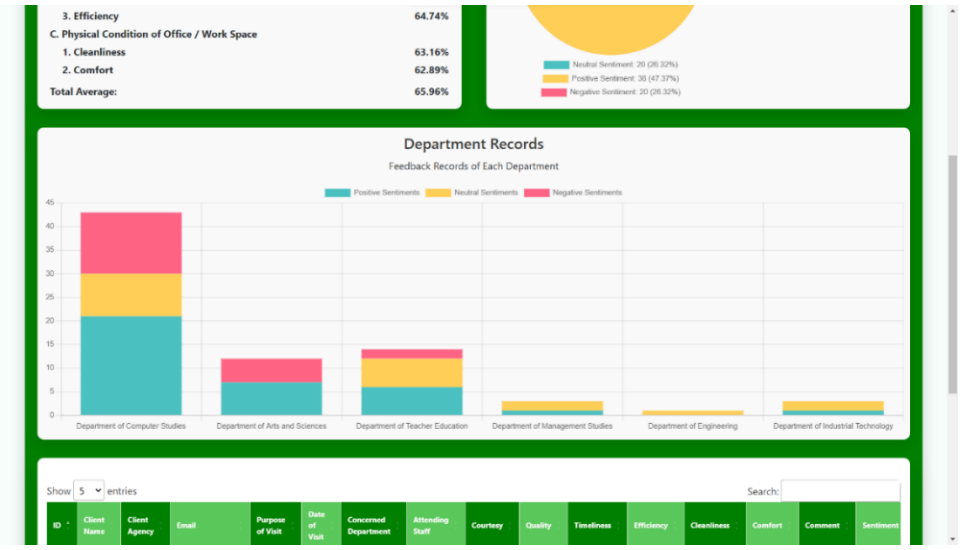


Figure 4-10 Admin department records for sentiment

Figure 4-11 shows the feedback of the client page. This page presents a comprehensive compilation of client feedback, enabling administrators to efficiently retrieve specific information through a search functionality. This feature allows administrators to streamline the process of obtaining relevant data by searching for specific keywords within the feedback dataset.

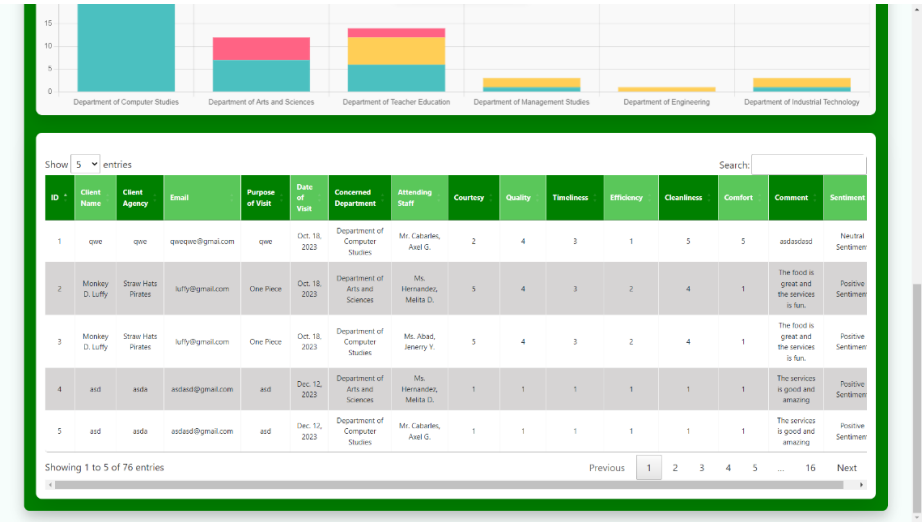


Figure 4-11 Admin client feedback copied form

Figure 4-12 shows the admin option page. The system provides administrative functionalities, encompassing options such as exporting data to collate results for sentiment analysis submissions, managing staff details by facilitating edits to personnel information within various offices, and offering a logout option for the seamless exit from the system.

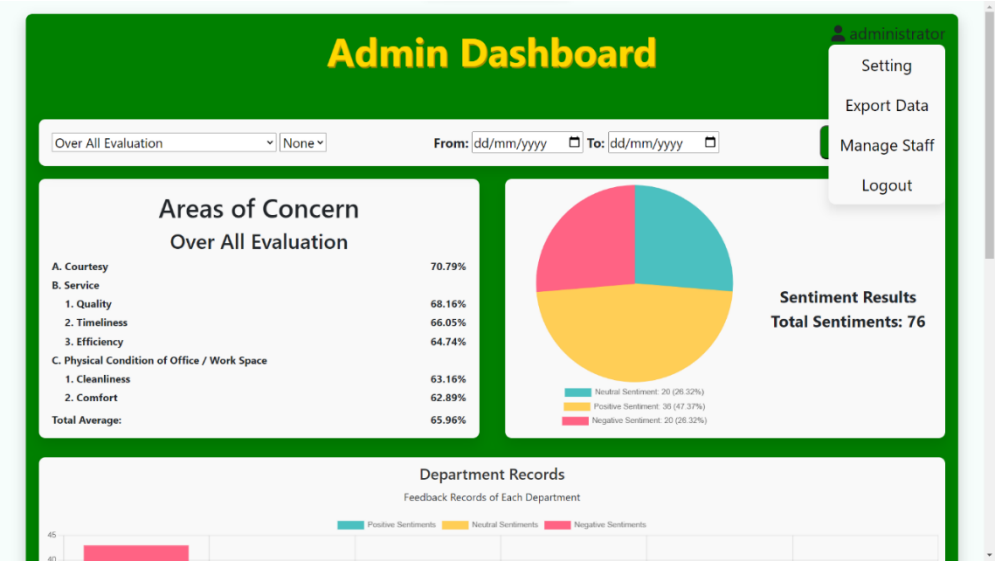


Figure 4-12 Admin option page

Figure 4-13 shows the admin setting page. This page allows administrators to enhance security by providing the capability to modify their username or password as needed. This functionality empowers administrators to proactively manage and fortify the security of their accounts within the system.

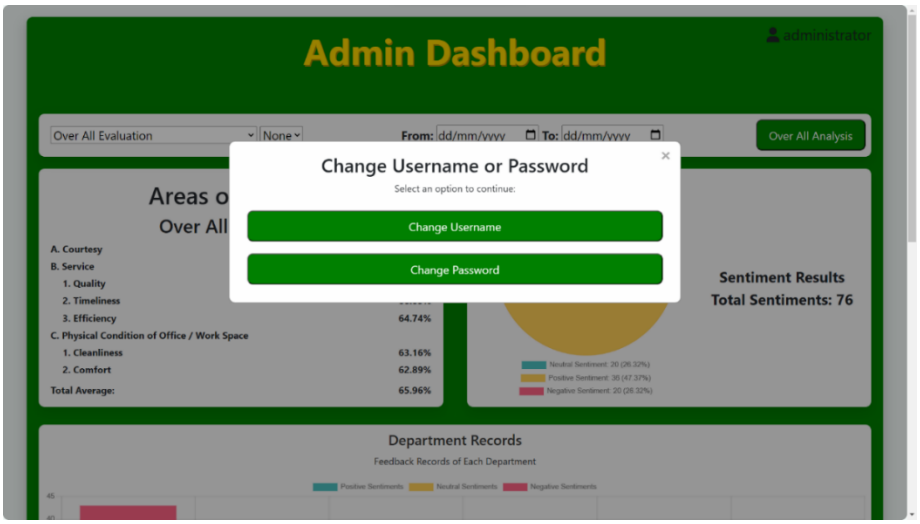


Figure 4-13 Admin setting page

Figure 4-14 shows the admin manage staff option page. This show administrator to incorporate, revise, and refresh the particulars pertaining to staff members across various offices at CvSU-CCAT Campus. This functionality serves as a valuable tool, particularly in the event of official updates to the staff roster, ensuring that the system remains current by aligning with the most recent information regarding personnel within each office.

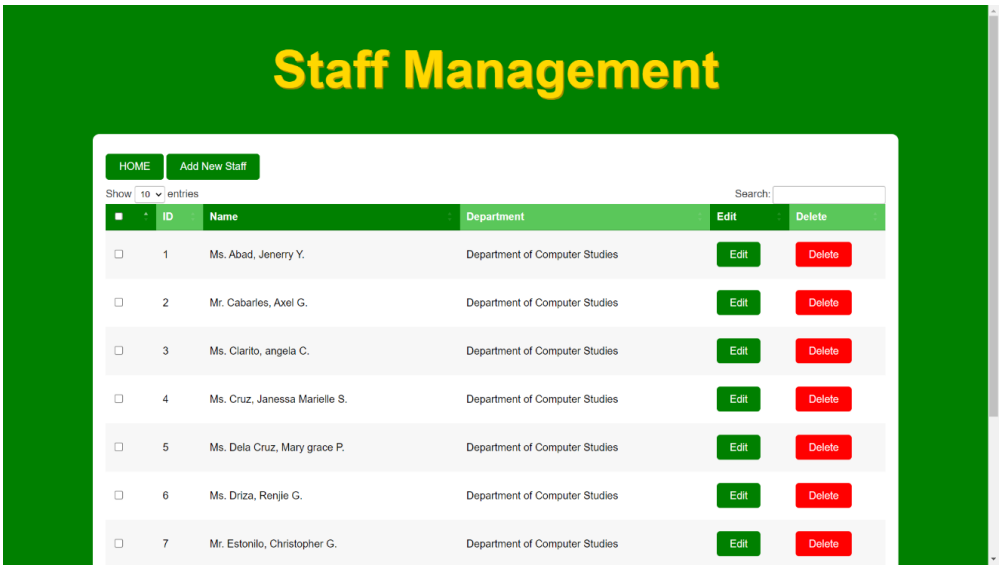


Figure 4-14 Admin staff management page

4.3 Results of the application evaluation

4.3.1 Functional Testing

Table 4-1 The summary of Functional Testing

Module	**Account**				
Test Case	**Action**	**Expected System Response**	**Number of Execution**	**Results**	**Status**
Feedback form Submit	Submit registration	Receives feedback record and show appreciation message	1 second after submission	Receives feedback record and show appreciation message	Passed
Login account	Submit login request	Display dashboard according to login credential	1 second after hitting login	Validates login credential and proceeds to dashboard.	Passed
Dashboard Staff, Department, and Date Filtering	Query feedback data or records based on the inputted criteria	Show feedback records according to criteria	1 second after hitting custom analysis button	Filter displayed data according to criteria	Passed
Search	Query data based on the inputted text	Show data based on the inputted text	Real-time	Display data related to inputted words	Passed
Pie chart and Bar graph Sentiment Filtering	Filter sentiments	Change the pie chart and bar graph data	Real-time	Pie chart and bar graph data according to selected legends	Passed

Table 4-1 contd.

Change password	Submit Change password request	There will be a successful message pop up and request is accepted	1 second	The system accepts the request and display pop up message	Passed
Change username	Submit change username request	There will be a pop up successful message and request is accepted	1 second	The system accepts the request and display pop up message	Passed
Export data	Export data based on the displayed records	Export file containing the data	1 sec after hitting export data	Export .txt file containing the data that is currently shown in the dashboard	Passed
Add new staff	Submit add new staff request	Staff information added to the database	1 second	New staff information added in the database.	Passed
Edit staff	Submit edit new staff request	Staff information update to the database	1 second	Staff information is updated.	Passed
Delete staff	Submit delete request	Delete staff	1 second	Staff Deleted	Passed
Logout	Submit logout request	Logout the user and end the session	1 second after hitting login	Logout the accounts	Passed

Table 4-14 shows the summary of functional testing of VISITHOUGHTS. The indicators were rated as follows: Feedback form Submit, Login account, Dashboard Staff, Department, and Date Filtering, Search, Pie chart and Bar graph Sentiment Filtering, change password, Change username, Export data, Add new staff, Edit staff, Delete staff, Logout, passed.

4.3.2 Software Evaluation from the End Users

Table 4-2 Perception of the end user on the usability of the website

USABILITY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Appropriateness Recognizability	4.76	Excellent
2. Learnability	4.76	Excellent
3. Operability	4.76	Excellent
4. User error protection	4.5	Excellent
5. User interface aesthetics	4.69	Excellent
TOTAL	**4.69**	**EXCELLENT**

Table 4-2 shows the perception of the respondents on usability of the website. The indicators were scored as follows: appropriateness recognizability, 4.76; learnability, 4.76; operability, 4.76; user error protection, 4.5; and user interface aesthetics, 4.69. The total average score is 4.69. It indicates that users can easily recognize the website based on their specific needs, promoting a cohesive grasp of its functionalities and supporting uncomplicated navigation and control.

Table 4-3 Perception of the end users on the effectiveness of the website

EFFECTIVENESS	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Effectiveness	4.76	Excellent
TOTAL	**4.76**	**EXCELLENT**

The perceptions of the respondents in the effectiveness of the system were shown in Table 4-3. The total score of 4.76 indicates that the website successfully met its objectives when utilized in particular contexts.

Table 4-4 Perception of the end users on the efficiency of the website

EFFICIENCY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Efficiency	4.73	Excellent
TOTAL	**4.73**	**EXCELLENT**

The perceptions of the respondents in the efficiency of the system were shown in Table 4-4. It gets a total score of 4.73 indicates that the website effectively and thoroughly assisted users with precision.

Table 4-5 Perception of the end users on the satisfaction of the website

SATISFACTION	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Usefulness	4.69	Excellent
2. Trust	4.50	Excellent
3. Pleasure	4.65	Excellent
4. Comfort	4.61	Excellent
TOTAL	**4.61**	**EXCELLENT**

The perception of the respondents in the satisfaction of the system was shown in Table 4-5. The indicators were rated as follows: usefulness, 4.69; trust, 4.5; pleasure, 4.65; and comfort, 4.61. The cumulative score of 4.61 suggests that the website was considered valuable, reliable, and adept at satisfying the physical comfort and personal requirements of its users.

Table 4-6 Perception of the end users on the context coverage of the website

CONTEXT COVERAGE	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Content completeness	4.69	Excellent
2. Flexibility	4.69	Excellent
TOTAL	**4.69**	**EXCELLENT**

The perception of the respondents in the context coverage of the website was shown in Table 4-6. The content completeness achieved a score of 4.69, and concurrently, the flexibility of the website also attained a score of 4.69. This confirms that the website has effectively fulfilled the requirements for both extensive content coverage and operational adaptability.

Table 4-7 Summary of results of the website evaluation assessed by the end users

CRITERIA	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Usability	4.69	Excellent
2. Effectiveness	4.76	Excellent
3. Efficiency	4.73	Excellent
4. Satisfaction	4.61	Excellent
5. Context Coverage	4.69	Excellent
TOTAL	**4.69**	**EXCELLENT**

The summary of the results of the evaluation of the website assessed by the end users was shown in Table 4-7. In general, survey participants conveyed a positive view of the website, describing it as easy to use, productive, efficient, and adept at fulfilling user requirements with thorough content. As a result, the website received an overall score of 4.69, indicating an Excellent rating.

4.3.3 Software Evaluation from the Web Developers

Table 4-8 Perception of the IT experts on the functionality of the website

FUNCTIONAL SUITABILITY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Functional completeness	4.80	Excellent
2. Functional correctness	4.80	Excellent
3. Functional appropriateness	5.00	Excellent
TOTAL	**4.93**	**EXCELLENT**

The perception of the respondents regarding the functional suitability of the website was reflected in Table 4-8. The indicators were rated as follows: functional completeness, 4.80; functional correctness, 4.80; and functional appropriateness, 5.00. The total score of 4.93 suggested that the website's intent was precise, suitable, and completely operational.

Table 4-9 Perceptions of the IT experts on the performance of the website

PERFORMANCE EFFICIENCY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Time behavior	5.00	Excellent
2. Resource utilization	5.00	Excellent
3. Capacity	5.00	Excellent
TOTAL	**5.00**	**EXCELLENT**

The perception of the respondents regarding performance efficiency was shown in Table 4-9. The indicators were scored as follows: time behavior, 5.00; resource utilization, 5.00; and capacity, 5.00. The total score of 5.00 Indicates that the website's purpose was effective in terms of performance.

Table 4-10 Perception of the IT experts on the usability of the website

USABILITY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Appropriateness recognizability	4.60	Excellent
2. Learnability	4.80	Excellent

Table 4-10 contd.

3. Operability	4.80	Excellent
4. User error protection	4.60	Excellent
5. User interface aesthetics	4.20	Very Satisfactory
6. Accessibility	4.80	Excellent
TOTAL	**4.63**	**EXCELLENT**

The perception of the respondents' regarding usability was shown in Table 4-10. The indicators were rated as follows: appropriateness recognizability, 4.60; learnability, 4.80; operability, 4.80; user error protection, 4.60; user interface aesthetics, 4.20; and accessibility, 4.80. The mean of 4.63 suggests that the website's intent was easily understandable, free from errors, and visually appealing to users.

Table 4-11 Perception of the respondents on security

SECURITY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Confidentiality	4.80	Excellent
2. Integrity	4.80	Excellent
3. Non-repudiation	4.80	Excellent
4. Accountability	4.80	Excellent
5. Authenticity	4.80	Excellent
TOTAL	**4.80**	**EXCELLENT**

The perception of the respondents' regarding security was shown in Table 4-11. The indicators were rated correspondingly: confidentiality, 4.80; integrity, 4.80; non-repudiation, 4.80; accountability, 4.80; and authenticity, 4.80. The score of 4.80 suggests

that the website safeguards against unauthorized access, ensuring that only users with granted permission can access the data.

Table 4-12 Perception of the IT experts on the maintainability of the website

MAINTAINABILITY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Modularity	4.80	Excellent
2. Reusability	5.00	Excellent
3. Analyzability	5.00	Excellent
4. Modification	5.00	Excellent
5. Testability	5.00	Excellent
TOTAL	**4.96**	**EXCELLENT**

The perception of the respondents' regarding maintainability was shown in Table 4-12. The indicators were rated as follows: modularity, 4.80; reusability, 5.00; analyzability, 5.00; modifiability, 5.00; testability, 5.00. The mean of 4.96 suggests that users with permission can effectively and efficiently modify the website's purpose without causing harm to the system.

Table 4-13 Perception of the IT experts on the portability of the website

PORTABILITY	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Adaptability	5.00	Excellent
TOTAL	**5.00**	**EXCELLENT**

The perception of the respondents' regarding portability was shown in Table 4-13. The indicators were rated as follows: adaptability, 5.00. The mean of 5.00 suggests that the website has the capability to adjust to various devices or browsers it was accessed on.

Table 4-14 Summary of results of the website evaluation assessed by the IT experts

CRITERIA	AVERAGE SCORE	DESCRIPTIVE INTEPRETATION
1. Functional Suitability	4.93	Excellent
2. Performance Efficiency	5.00	Excellent
3. Usability	4.63	Excellent
4. Security	4.80	Excellent
5. Maintainability	4.96	Excellent
6. Portability	5.00	Excellent
TOTAL	**4.88**	**EXCELLENT**

Table 4-14 shows the overall summary of the results of the website evaluation assessed by the IT experts. Consequently, the website scored a total of 4.88, described as Excellent.

5 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

5.1 Summary

The research spanned from June 2023 to December 2023 to streamline the feedback process at Cavite State University - CCAT Campus. This involved developing a web application to render the feedback form easily accessible, timely, and paperless for clients. The primary objective was to collect feedback efficiently and thoroughly analyze clients' evaluations and comments. Specifically, the study aims to: 1. identify the essential requirements and functionalities of the web application through: a) conducting a thorough analysis of the feedback collection method and analysis needs of various offices and units within Cavite State University CCAT Campus. b) identifying the essential features and functionalities that should be incorporated into the VisiThoughts web application to meet the identified needs. c)defining the user interface design and user experience guidelines to ensure a user-friendly and accessible platform. 2. develop the VisiThoughts web application using a Lexicon-based approach. 3. evaluate the application to determine if it complies with the modified ISO-IEC 25010 standards.4. implement the system in the CvSU-CCAT Campus.

The evaluation of the system involved 31 participants, including five (5) IT experts and twenty-six (26) end users. The assessment results revealed an impressive average score of 4.88 from the IT experts, characterized as "Excellent," and an average score of 4.69 from the end users, also interpreted as "Excellent." These findings affirmed that the developed web application system had demonstrated its capability to deliver satisfaction, reliability, and effectiveness in evaluation analysis.

5.2 Conclusions

The VisiThoughts web application, created for the CvSU-CCAT Campus to analyze customer comments using sentiment analysis, was proof of the efficient implementation of the specialized Natural Language Toolkit (NLTK) algorithm. This advanced system was a great help to visitors, department heads, and administrators because it made it easy to collect detailed input from several offices on the Cavite State University-CCAT campus. The incorporation of VisiThoughts has been crucial in improving office services by showcasing the outcomes of comprehensive assessments, efficiently gathering client input, and emphasizing remarks. The flexibility provided to clients, which allowed them to provide feedback via mobile phones or the central system using the university's standardized feedback form, improves the system's accessibility. VisiThoughts findings contribute considerably to the continued refinement of services, but they were also personalized for submission and presentation to each separate office, promoting a continual cycle of improvement.

VisiThoughts stressed user-friendliness by incorporating interactive language, simplifying evaluators' interpretation of the feedback form. The slider tool simplifies the procedure further, allowing evaluators to complete the form quickly. Furthermore, the system emphasized data security by requiring users to agree to terms and conditions, reducing potential data violations, and assuring a secure user experience. VisiThoughts, in general, achieves its study objectives and emerges as a potent tool for ongoing service enhancement at the CvSU-CCAT Campus. The study represented significant progress in the continuous improvement of services within an educational institution and a substantial contribution to interactive systems and evaluation analysis.

5.3 Recommendations

In light of the evaluation feedback received from both end users and IT experts, the following recommendations are proposed:

1. Enhance the clarity and specificity of the terms and conditions.
2. Implement access limitations to the form to bolster security measures.
3. Provide comprehensive instructions throughout all sections of the form, with particular emphasis on sliders for improved user guidance.

6 REFERENCES

Alade, Samuel Mayowa, Adejumo, Samuel, Alade, Temitope John. (2022). Retrieved from https://ijcat.com/archieve/volume11/issue4/ijcatr11041006.pdf

Alejandra Segura Navarrete, Claudia Martinez-Araneda, Christian Vidal-Castro, Clemente Rubio-Manzano. (2021). Retrieved from https://www.emerald.com/insight/content/doi/10.1108/EL-04-2020-0110/full/html

Amal Htait, Leif A. Azzopardi. (2021). Retrieved from https://dl.acm.org/doi/10.1145/3471158.3472254

Amrutha Mohanan. K. V, Hema. E. K, Monisree. S, Audline Beena. S. P. (2019). Retrieved from https://www.ijtsrd.com/engineering/computer-engineering/11240/raspberry-pibot-manipulated-with-skid-steered-mechanism-for-multi-environment-surveillance/amrutha-mohanan-k-v

Anirban Adak, Biswajeet Pradhan, and Nagesh Shukla . (2022). Retrieved from https://www.mdpi.com/2304-8158/11/10/1500

Ara Rivina N. Malaya, Emely A. Munar, Floribeth Panay Cuison, Enrico G. Dacanay. (2022). Retrieved from https://ijeecs.iaescore.com/index.php/IJEECS/article/view/30145

Bhargav N, Dr. Jayanthi P N. (2021). Retrieved from https://jusst.org/real-time-office-seat-allocation-system-using-spring-mvc-framework/

Cailing Wang, Yiming Wang, Jing Zhang. (2023). Retrieved from https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12635/2679069/Development-and-deployment-of-a-responsive-library-platform-based-on/10.1117/12.2679069.short?SSO=1

Chayanan Kerdpitak. (2021). Retrieved from http://jsju.org/index.php/journal/article/view/879

DEVI, D. G. (2023). Retrieved from http://magesticts.com/13.html

DHANALAKSHMI G, DAPHNE PATRICIA P, ANU SOWMIYAA A, ARUNA RAJESWARI K K. (2023). Retrieved from https://journals.innovareacademics.in/index.php/ijet/article/view/47235

Ertie Abana, Benedict Sy. (2022). Retrieved from https://ijeecs.iaescore.com/index.php/IJEECS/article/view/29103

Farhad Mofidi, Hashem Akbari. (2019). Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0378778818319704?via%3Dihub

Fersini, E. (2017). Retrieved from https://www.sciencedirect.com/topics/computer-science/sentiment-orientation

Francisco Villarroel Ordenes, Babis Theodoulidis, Jamie Burton, Thorsten Gruber, Mohamed Zaki. (2014). Retrieved from https://journals.sagepub.com/doi/10.1177/1094670514524625

I Gede Surya Rahayuda, Ni Putu Linda Santiari. (2021). Retrieved from http://openjournal.unpam.ac.id/index.php/informatika/article/view/10077

Jose Ramon Saura, B. R.-M. (2019). Retrieved from https://ieeexplore.ieee.org/document/8667819

Jose Ramon Saura, Felipe Debasa, Ana Reyes-Menendez. (2019). Retrieved from Jose Ramon Saura 1, Felipe Debasa 2, Ana Reyes-Menendez

Karpei & Kormuzhanin, 2021. (2021). Retrieved from https://disk.yandex.by/i/I7Eipk7W_l_c5Q

Kartik Bajaj, Karthik Pattabiraman, Ali Mesbah. (2014). Retrieved from https://dl.acm.org/doi/10.1145/2597073.2597083

Kayıkçı, Ş. (2022). Retrieved from https://link.springer.com/article/10.1007/s11042-022-11929-w

Kozuyla & Sushko. (2021). Retrieved from http://samit.khpi.edu.ua/article/view/249608

Li Yang, Ying Li, Jin Wang, R. Simon Sherratt, . (2020). Retrieved from https://ieeexplore.ieee.org/document/8970492

Long Hoang Nguyen, Bach Xuan Tran, Luis E. C. Rocha, Huong Lan Thi Nguyen, Cui Yang, Carl A. Latkin, Anna Thorson & Susanne Strömdahl . (2019). Retrieved from https://link.springer.com/article/10.1007/s10461-019-02626-1

Madiha Khalid, Imran Ashraf, Arif Mehmood, Saleem Ullah, Maqsood Ahmad, and Gyu Sang Choi. (2020). Retrieved from https://www.mdpi.com/2076-3417/10/8/2788

Manjula Selvam, M. Ramachandran, Prabakaran Nanjundan, Malarvizhi Mani. (2022). Retrieved from http://restpublisher.com/wp-content/uploads/2023/01/10.46632-jitl-1-1-3.pdf

Marina Obradović , Fakultet organizacijskih znanosti, Sveučilište u Beogradu, Beograd, Srbija. (2022). Retrieved from https://hrcak.srce.hr/clanak/416501

Marouen Ben Guebila, Camila M Lopes-Ramos, Deborah Weighill, Abhijeet Rajendra Sonawane, Rebekka Burkholz, Behrouz Shamsaei, John Platig, Kimberly Glass, Marieke L Kuijjer, John Quackenbush. (2022). Retrieved from https://disk.yandex.by/i/I7Eipk7W_l_c5Q

Michelle Lobchuk, Lisa Hoplock; Nicole Harder, Marcia Friesen, Julie Rempel, Prachotan Reddy Bathi,. (2023). Retrieved from https://formative.jmir.org/2023/1/e41222

Mohammed Al Ani, George Garas, James Hollingshead, Drostan Cheetham, Thanos Athanasiou, Vanash Patel. (2022). Retrieved from https://karger.com/mpp/article/31/4/342/825089/Which-Electronic-Health-Record-System-Should-We

Muhammad Zubair Asghar, Ikram Ullah , Shahab Shamshirband, Fazal Masud Kundi , Ammara Habib. (2019). Retrieved from https://www.preprints.org/manuscript/201907.0006/v1

N L P Suciptawati, N L P S P Paramita, and I P Aristayasa. (2019). Retrieved from https://iopscience.iop.org/article/10.1088/1742-6596/1321/2/022055/pdf

Neha Gupta, Rashmi Agrawal. (2020). Retrieved from https://www.sciencedirect.com/topics/computer-science/sentiment-orientation

Paulo Rita, Celeste Vong, Flávio Pinheiro, João Mimoso. (2022). Retrieved from (https://www.emerald.com/insight/content/doi/10.1108/EJMBE-11-2021-0295/full/html

Philip Lew, Luis Olsina & Li Zhang . (2010). Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-13911-6_15

Rai, Z. S. (2022). Retrieved from https://ijarsct.co.in/Paper5743.pdf

Ramesh Chundi, Vishwanath R. Hulipalled, Jay Bharthish Simha. (2023). Retrieved from https://ijai.iaescore.com/index.php/IJAI/article/view/22559

Reda Alkot, Sayed Ahmed Sayed. (2022). Retrieved from https://bsge.journals.ekb.eg/article_284700.html

Rossano Atzeni, Matteo Massidda, Giorgio Fotia, Paolo Uva. (2022). Retrieved from https://onlinelibrary.wiley.com/doi/10.1002/humu.24495

Saeed-Ul Hassan, Aneela Saleem, Saira Hanif Soroya, Iqra Safder, Sehrish Iqbal, Saqib Jamil, Faisal Bukhari, Naif Radi Aljohani, Raheel Nawaz. (2020). Retrieved from https://journals.sagepub.com/doi/10.1177/0165551520930917

Thiemo Wambsganss, Naim Zierau, Matthias Söllner, Tanja Käser, Kenneth R Koedinger, J. M. Leimeister. (2022). Retrieved from https://dl.acm.org/doi/10.1145/3555619

Wenjing Duan, Yang Yu, Qing Cao, Stuart Levy. (2015). Retrieved from https://journals.sagepub.com/doi/10.1177/193896551562048

APPENDICES

APPENDIX A
ROUTING SLIP

Republic of the Philippines
CAVITE STATE UNIVERSITY
CCAT Campus
Rosario, Cavite
☏ (046) 437-9505 / 🖷 (046) 437-6659
cvsurosario@cvsu.edu.ph
www.cvsu-rosario.edu.ph

DEPARTMENT OF COMPUTER STUDIES

ROUTING SLIP FOR FINAL MANUSCRIPT

Name: : **ELGENE JOHN V. REYES**
KARL DOHNEYZEL A. DELACRUZ
JAN PAOLO A. CANO

Degree//Specialization : **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

Study Title: : **VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS**

Type of Study : ● Thesis ○ EDP ○ Design Project ○ Case Study ○ Capstone

NAME	DATE		REMARKS
	RECEIVED	RELEASED	
CHRISTOPHER G. ESTONILO, DIT *Research Adviser*	1/18/2024 1/24/2024 2/6/2024	1/22/2024 1/30/2024 2/8/2024	
JEFFRID D. LAXAMANA *Technical Critic*			
JENERRY Y. ABAD *Unit Research Coordinator*	02/01/2024 02/08/2024	02/01/2024 02/08/2024	
KARLO JOSE E. NABABLIT, MSICT *Department Chairperson*	02-08-24	02-08-24	ok
RENATO A. BAUTISTA JR. *English Critic*	01-30-24 02-06-24	02-02-24 02-08-24	
ARIES M. GELERA, MSICT *Director, Research and Extension*	02-12-24	02-12-24	ok
ARIEL G. SANTOS, EdD *Director for Instruction*			
LAURO B. PASCUA, EdD *Campus Administrator*		02.12.24	

APPENDIX B
REQUEST FOR ADVISER AND TECHNICAL CRITIC

Republic of the Philippines
CAVITE STATE UNIVERSITY
CCAT Campus
Rosario, Cavite
(046) 437-9505 / (046) 437-6659
cvsurosario@cvsu.edu.ph
www.cvsu-rosario.edu.ph

DEPARTMENT OF COMPUTER STUDIES

REQUEST FOR ADVISER AND TECHNICAL CRITIC

Name:	:	**REYES , ELGENE JOHN V.** **CANO , JAN PAOLO A.** **DELA CRUZ , KARL DOHNEYZEL A.**
Degree//Specialization	:	**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**
Study Title:	:	**"LeagueTas Map": The Development Evacuation Assistant Mobile Application. Built with Global Positioning System (GPS) and Natural Language Processing (NLP)**

Conforme:

CHRISTOPHER G. ESTONILO, DIT 3/27/2023 **JEFFRID D. LAXAMANA** _____
Adviser Date Technical Critic Date

Approved:

KARLO JOSE E. NABABLIT, MSICT 3/27/23 **ARIES M. GELERA, MSICT** 03-28-23
Unit Research Coordinator Date Department Chairperson Date

Form No.	Revision Date	Effectivity Date
CvSU-CCAT SR Form No. 1A	13 August 2018	10 October 2018

APPROVAL C
REQUEST FOR OUTLINE DEFENSE

Republic of the Philippines
CAVITE STATE UNIVERSITY
Cavite College of Arts and Trades
CvSU-CCAT Campus
Rosario, Cavite
☎ (046) 437-9505 / 🖷 (046) 437-6659
cvsu.rosario@gmail.com
www.cvsu-rosario.edu.ph

DEPARTMENT OF COMPUTER STUDIES

July 02, 2023

ARIES M. GELERA
Chairperson
Department of Computer Studies

Sir:

We wish to apply for an oral review of our thesis outline entitled **Visithoughts: A Web Application for Customers Feedback with Sentiment Analysis of Cvsu-Ccat Campus** on **July 20223** at the Audio-visual Room of the Department of Computer Studies.

Thank you.

Sincerely yours,

ELGENE JOHN V. REYES
KARL DHONHEYZEL A. DELA CRUZ
JAN PAOLO A. CANO

Recommending Approval:

CHRISTOPHER G. ESTONILO,DIT 7/10/2023
Adviser Date

JEFERID D. LAXAMANA _____
Technical Critic Date

Noted:

KARLO JOSE E. NABABLIT, MSICT _____
Unit Research Coordinator Date

Approved by:

ARIES M. GELERA, MSICT 7/10/23
Chairperson Date

Form No.	Revision Date	Effectivity Date
CvSU-CCAT SR Form No. 3	13 August 2018	10 October 2018

APPROVAL D
REQUEST FOR FINAL DEFENSE

Republic of the Philippines
CAVITE STATE UNIVERSITY
CCAT Campus
Rosario, Cavite
☎ (046) 437-9505 / 🖷 (046) 437-6659
cvsurosario@cvsu.edu.ph
www.cvsu-rosario.edu.ph

DEPARTMENT OF COMPUTER STUDIES

November 28, 2023

DR. LAURO B. PASCUA
Campus Administrator
This Campus

Sir:

We wish to apply for a final review of our thesis manuscript entitled **VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS** on **December 6, 2023** at the Central Computer Laboratory of the Department of Computer Studies.

Thank you.

Sincerely yours,

ELGENE JOHN V. REYES
KARL DOHNEYZEL A. DELACRUZ
JAN PAOLO A. CANO

Recommending Approval:

CHRISTOPHER G. ESTONILO, DIT 11/28/2023
Adviser — Date

JEFFRID D. LAXAMANA 11/28/23
Technical Critic — Date

JENERRY Y. ABAD 11/28/2023
Unit Research Coordinator — Date

KARLO JOSE E. NABABLIT, MSICT 11/28/23
Department Chairperson — Date

ARIEL G. SANTOS, EdD 11/29/2023
Director, Instructions — Date

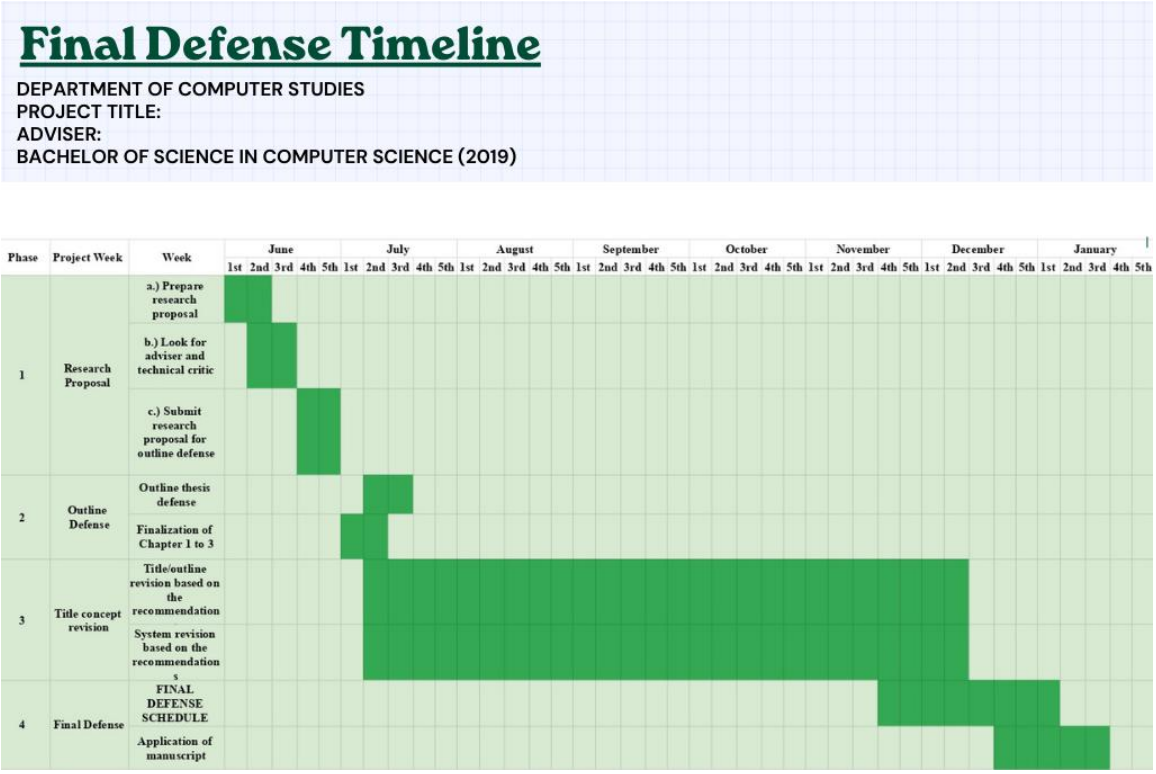
Approved:

LAURO B. PASCUA, EdD 11/29/2023
Campus Administrator — Date

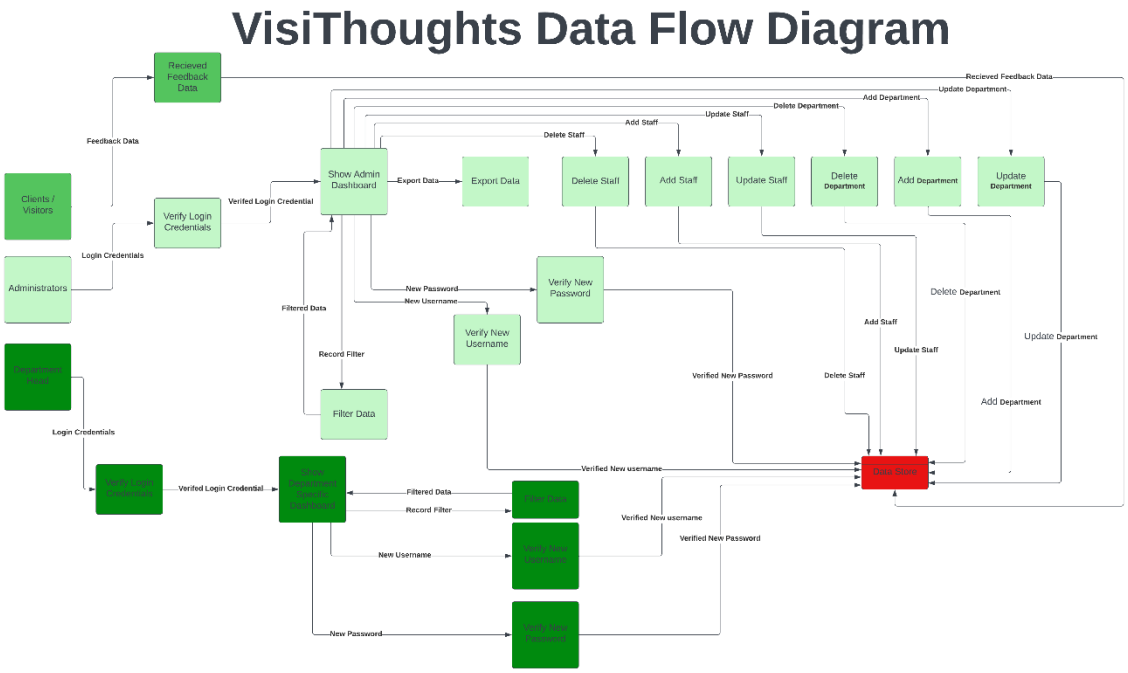
Form No.	Revision Date	Effectivity Date
CvSU-CCAT SR Form No.6	13 August 2018	10 October 2018

APPENDIX E
CERTIFICATION OF ENGLISH CRITIC

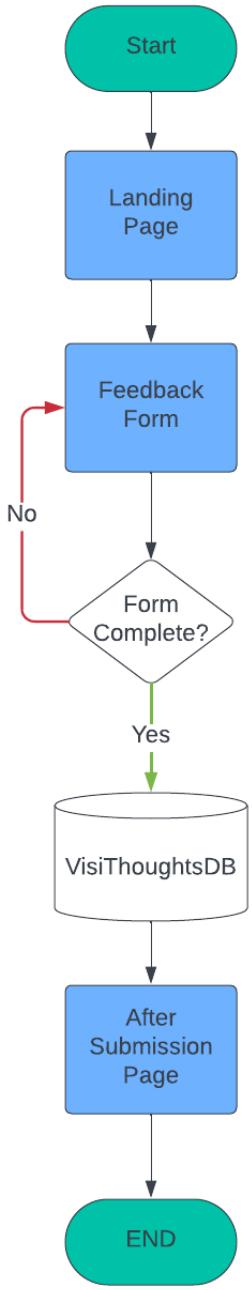
APPENDIX F
OTHER APPENDICES



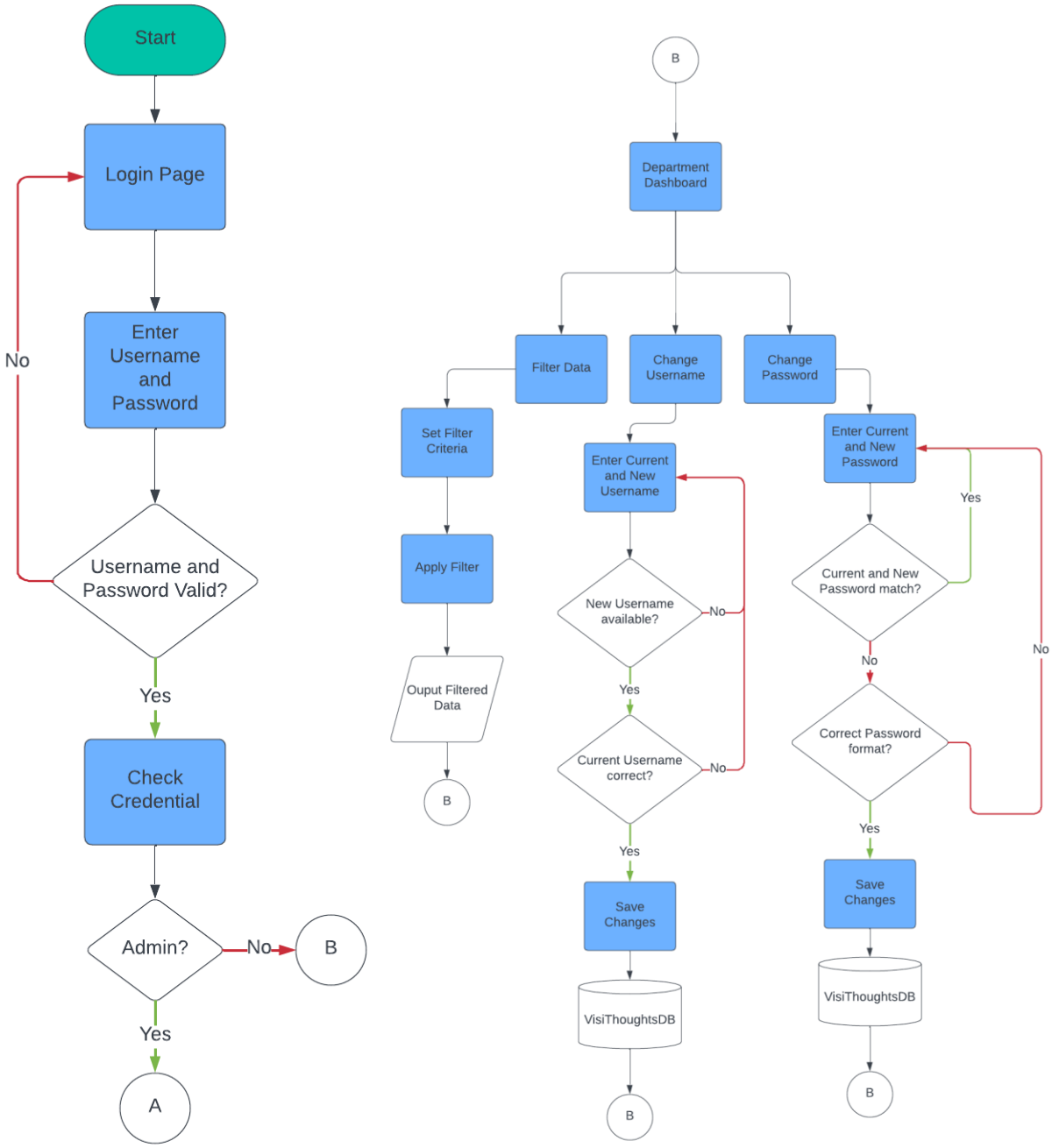
Gantt Chart



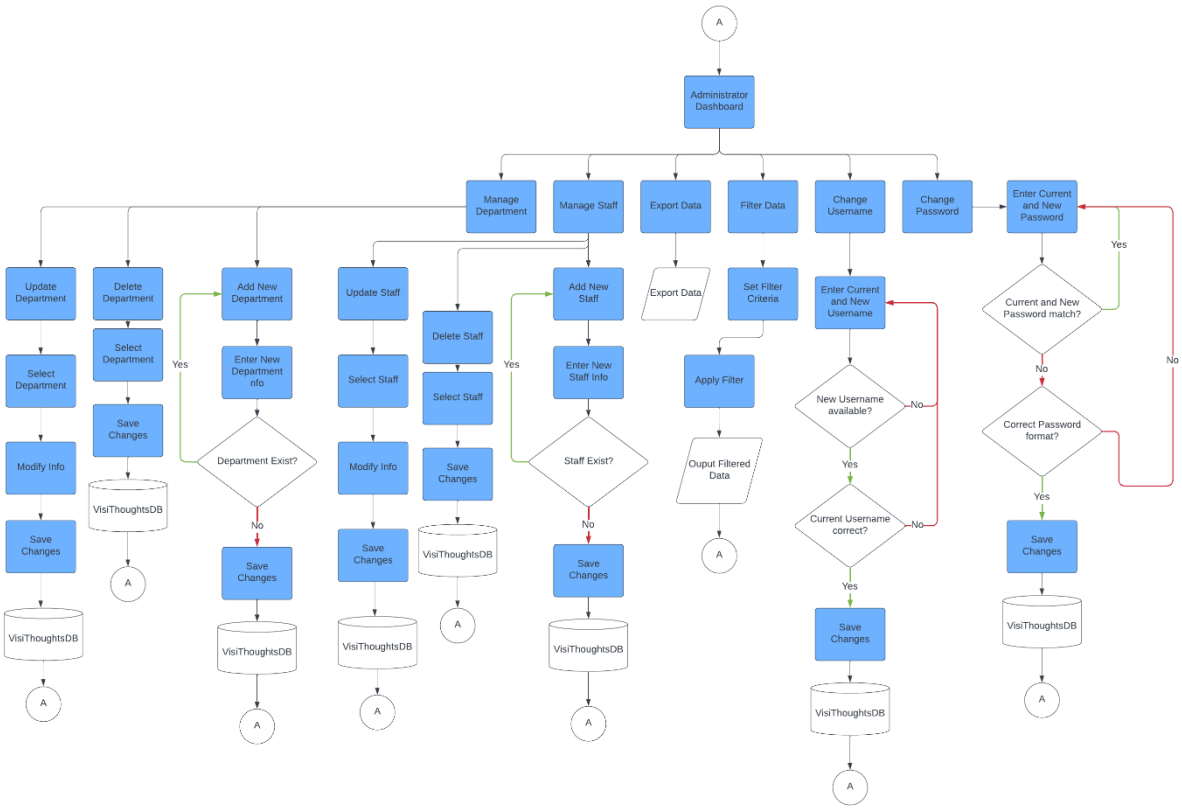
Data Flow Diagram



Flow Chart for End-Users



Flow Chart for Administrator and Department Head



Flow Chart for Administrator and Department Head

APPENDIX G
EVALUATION OF DEVELOPERS

DCS-SEI Form No. 1

Republic of the Philippines
CAVITE STATE UNIVERSITY
Cavite College of Arts and Trades
CvSU-CCAT Campus
Rosario, Cavite

SOFTWARE EVALUATION INSTRUMENT FOR
VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS
(For Web Developers)

To the respondents,

The primary objective of this software evaluation is to assess your perception on the different technical and non-technical characteristics of the software material. As a participant, your responses to this instrument shall signify for your consent to disclose your opinions and other information relevant to the requirements of the instrument, and the evaluation in general. Rest assured that as a participant, you will be given the highest level of anonymity and any personal information that will be disclosed will be treated with confidentiality. All the information to be gathered shall solely serve educational purposes.

Name of Respondent: RENJIE G. DRIZA **Age:** 25 **Gender:** ( ) M (✓) F
Profession/Occupation INSTRUCTOR **Affiliation:** _____

General Instruction: Evaluate the characteristics of the web application using the rating scale below. Place a checkmark (✓) on the corresponding numerical rating.

Rating Scale

NUMERICAL RATING	DESCRIPTION
5	Excellent
4	Very Satisfactory
3	Satisfactory
2	Unsatisfactory
1	Needs Improvement

INDICATOR	5	4	3	2	1
A. Functional Suitability					
1. Functional completeness - degree to which the set of functions covers all the specified tasks and user objectives	✓				
2. Functional correctness - degree to which the functions provide the correct results with precision	✓				
3. Functional appropriateness - degree to which the functions facilitate the accomplishment of specified tasks and objectives	✓				

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

DCS-SEI Form No. 1

INDICATOR	5	4	3	2	1
B. Performance Efficiency					
1. Time behavior - degree to which the responses, processing times, and throughput rates of the website meet the requirements when performing its functions	✓				
2. Resource utilization - degree to which the amounts and the types of resources used by the website meet the requirements when performing its functions	✓				
3. Capacity - degree to which the maximum limitations of the website or its components meet the requirements	✓				
C. Usability					
1. Appropriateness recognizability - degree to which users recognize the appropriateness of the website based on their needs		✓			
2. Learnability - degree to which the website enables the user to learn how to use the software effectively	✓				
3. Operability - degree to which the website is easy to operate, control, and be used appropriately	✓				
4. User error protection - degree to which the website protects the user from making errors		✓			
5. User interface aesthetics - degree to which the user interface enables pleasing and satisfying interaction with the user		✓			
6. Accessibility - degree to which the website can be used in a specific context with the widest range of characteristics and capabilities to achieve a specific goal	✓				
D. Security					
1. Confidentiality - degree to which the website ensures that the data are only accessible by those who have authorized access		✓			
2. Integrity - degree to which the website prevents unauthorized access to and or modifications of computer programs or data		✓			
3. Non-repudiation - degree to which actions or events can be proven to have taken place to avoid repudiation later on		✓			
4. Accountability - degree to which the actions of an entity can be uniquely accounted to the entity		✓			
5. Authenticity - degree to which the identity of a subject or resource can be proven to be the one being claimed		✓			
E. Maintainability					
1. Modularity - degree to which the website consists of discrete components so that a change on one component pose minimal impact to other components		✓			
2. Reusability - degree to which an asset can be used in more than one system or in building other assets	✓				
3. Analyzability - degree of effectiveness and efficiency on possibly assessing the impact of an intended change to one or more parts of the website; diagnosing deficiencies or causes of failures; and or	✓				

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

DCS-SEI Form No. 1

identifying parts of the website to be modified					
INDICATOR	**5**	**4**	**3**	**2**	**1**
E. Maintainability					
4. Modifiability - degree to which the website can be effectively and efficiently modified without introducing defects or degrading existing quality	✓				
5. Testability - degree in which test criteria can be efficiently and effectively established and the software or components can be tested to determine whether test criteria were met	✓				
F. Portability					
1. Adaptability - degree to which the website can be effectively and efficiently adapted for different or evolving hardware, software, or other operational or usage environments.	✓				

Remarks:

+ more on password and authentication
+ Good JOB! Congrats! :)

Certified true and correct:

KENJIE G. DRIZA
Signature over printed name of evaluator

NOV. 22, 2023
Date

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

APPENDIX H
SAMPLE EVALUTION OF DEVELOPERS

DCS-SEI Form No. 1

Republic of the Philippines
CAVITE STATE UNIVERSITY
Cavite College of Arts and Trades
CvSU-CCAT Campus
Rosario, Cavite

SOFTWARE EVALUATION INSTRUMENT FOR
CCATA: AN ACCREDITATION INQUIRY CHATBOT APPLICATION USING NATURAL LANGUAGE PROCESSING FOR CAVITE STATE UNIVERSITY – CCAT CAMPUS
(For Web Developers)

To the respondents,

The primary objective of this software evaluation is to assess your perception on the different technical and non-technical characteristics of the software material. As a participant, your responses to this instrument shall signify for your consent to disclose your opinions and other information relevant to the requirements of the instrument, and the evaluation in general. Rest assured that as a participant, you will be given the highest level of anonymity and any personal information that will be disclosed will be treated with confidentiality. All the information to be gathered shall solely serve educational purposes.

Name of Respondent: _____ **Age:** _____ **Gender:** ( ) M ( ) F
Profession/Occupation _____ **Affiliation:** _____

General Instruction: Evaluate the characteristics of the web application using the rating scale below. Place a checkmark (✓) on the corresponding numerical rating.

Rating Scale

NUMERICAL RATING	DESCRIPTION
5	Excellent
4	Very Satisfactory
3	Satisfactory
2	Unsatisfactory
1	Needs Improvement

INDICATOR	5	4	3	2	1
A. Functional Suitability					
1. Functional completeness - degree to which the set of functions covers all the specified tasks and user objectives					
2. Functional correctness - degree to which the functions provide the correct results with precision					
3. Functional appropriateness - degree to which the functions					

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

facilitate the accomplishment of specified tasks and objectives					

INDICATOR	5	4	3	2	1
B. Performance Efficiency					
1. Time behavior - degree to which the responses, processing times, and throughput rates of the website meet the requirements when performing its functions					
2. Resource utilization - degree to which the amounts and the types of resources used by the website meet the requirements when performing its functions					
3. Capacity - degree to which the maximum limitations of the website or its components meet the requirements					
C. Usability					
1. Appropriateness recognizability - degree to which users recognize the appropriateness of the website based on their needs					
2. Learnability - degree to which the website enables the user to learn how to use the software effectively					
3. Operability - degree to which the website is easy to operate, control, and be used appropriately					
4. User error protection - degree to which the website protects the user from making errors					
5. User interface aesthetics - degree to which the user interface enables pleasing and satisfying interaction with the user					
6. Accessibility - degree to which the website can be used in a specific context with the widest range of characteristics and capabilities to achieve a specific goal					
D. Security					
1. Confidentiality - degree to which the website ensures that the data are only accessible by those who have authorized access					
2. Integrity - degree to which the website prevents unauthorized access to and or modifications of computer programs or data					
3. Non-repudiation - degree to which actions or events can be proven to have taken place to avoid repudiation later on					
4. Accountability - degree to which the actions of an entity can be uniquely accounted to the entity					
5. Authenticity - degree to which the identity of a subject or resource can be proven to be the one being claimed					
E. Maintainability					
1. Modularity - degree to which the website consists of discrete components so that a change on one component pose minimal impact to other components					
2. Reusability - degree to which an asset can be used in more than one system or in building other assets					
3. Analyzability - degree of effectiveness and efficiency on possibly					

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

DCS-SEI Form No. 1

assessing the impact of an intended change to one or more parts of the website; diagnosing deficiencies or causes of failures; and or identifying parts of the website to be modified					
INDICATOR	**5**	**4**	**3**	**2**	**1**
E. Maintainability					
4. Modifiability - degree to which the website can be effectively and efficiently modified without introducing defects or degrading existing quality					
5. Testability - degree in which test criteria can be efficiently and effectively established and the software or components can be tested to determine whether test criteria were met					
F. Portability					
1. Adaptability - degree to which the website can be effectively and efficiently adapted for different or evolving hardware, software, or other operational or usage environments.					

Remarks:

_____

_____

_____

_____

Certified true and correct:

_____ _____

Signature over printed name of evaluator Date

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

APPENDIX I
EVALUATION FOR END USERS

DCS-SEI Form No. 2

Republic of the Philippines
CAVITE STATE UNIVERSITY
Cavite College of Arts and Trades
CvSU-CCAT Campus
Rosario, Cavite

SOFTWARE EVALUATION INSTRUMENT FOR
VISITHOUGHTS: A WEB APPLICATION FOR CUSTOMERS FEEDBACK WITH SENTIMENT ANALYSIS OF CvSU-CCAT CAMPUS
(For End-users)

To the respondents,

The primary objective of this software evaluation is to assess your perception on the different technical and non-technical characteristics of the software material. As a participant, your responses to this instrument shall signify for your consent to disclose your opinions and other information relevant to the requirements of the instrument, and the evaluation in general. Rest assured that as a participant, you will be given the highest level of anonymity and any personal information that will be disclosed will be treated with confidentiality. All the information to be gathered shall solely serve educational purposes.

Name of Respondent: Judilyn E. Gonzales **Age:** 25 **Gender:** ( ) M (✓) F
Profession/Occupation Instructor I **Affiliation:** DTE

General Instruction: Evaluate the characteristics of the web application using the rating scale below. Place a checkmark (✓) on the corresponding numerical rating.

Rating Scale

NUMERICAL RATING	DESCRIPTION
5	Excellent
4	Very Satisfactory
3	Satisfactory
2	Unsatisfactory
1	Needs Improvement

INDICATOR	5	4	3	2	1
A. Usability					
1. Appropriateness recognizability - degree to which users recognize the appropriateness of the website based on their needs	✓				
2. Learnability - degree to which the website provides self-explanatory user-interface; entry fields contain default/placeholder texts; and error messages provided are understandable	✓				
INDICATOR	**5**	**4**	**3**	**2**	**1**

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

DCS-SEI Form No. 2

A. Usability					
3. Operability - degree to which the website is easy to operate, control, and can be used appropriately	✓				
4. User error protection - the website is able to perform user entry validation and allows the user to recover from errors	✓				
5. User interface aesthetics - degree to which the user interface enables pleasing and satisfying interaction with the user	✓				
B. Effectiveness					
1. The website allows the users to achieve specified goals when used in specified contexts	✓				
C. Efficiency					
1. The degree to which resources are consumed with accuracy and completeness of helping users achieve the goals	✓				
D. Satisfaction					
1. Usefulness - degree to which the user is satisfied on perceived achievement of pragmatic goals including the results and consequences of using the website	✓				
2. Trust - degree to which a user or other stakeholders are confident that the website will behave as it is intended	✓				
3. Pleasure - degree to which a user gains pleasure upon fulfillment of personal needs	✓				
4. Comfort - degree to which the user is satisfied with physical comfort	✓				
E. Context Coverage					
1. Content completeness - degree to which the website can be used effectively, efficiently, and satisfactorily in all specified contexts of use	✓				
2. Flexibility - degree to which the website can be used effectively, efficiently, and satisfactorily in contexts beyond those initially specified in the requirements	✓				

Remarks:

Very useful and creative, looking forward for its implementation

Certified true and correct:

Judilyn E. Gonzales

Signature over printed name of evaluator

11/21/2023

Date

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

APPENDIX J
SAMPPLE EVALUATION FOR END USERS

DCS-SEI Form No. 2

Republic of the Philippines
CAVITE STATE UNIVERSITY
Cavite College of Arts and Trades
CvSU-CCAT Campus
Rosario, Cavite

SOFTWARE EVALUATION INSTRUMENT FOR
CCATA: AN ACCREDITATION INQUIRY CHATBOT APPLICATION USING NATURAL LANGUAGE PROCESSING FOR CAVITE STATE UNIVERSITY – CCAT CAMPUS
(For End-users)

To the respondents,

The primary objective of this software evaluation is to assess your perception on the different technical and non-technical characteristics of the software material. As a participant, your responses to this instrument shall signify for your consent to disclose your opinions and other information relevant to the requirements of the instrument, and the evaluation in general. Rest assured that as a participant, you will be given the highest level of anonymity and any personal information that will be disclosed will be treated with confidentiality. All the information to be gathered shall solely serve educational purposes.

Name of Respondent: _____ **Age:** _____ **Gender:** ( ) M ( ) F
Profession/Occupation _____ **Affiliation:** _____

General Instruction: Evaluate the characteristics of the web application using the rating scale below. Place a checkmark (✓) on the corresponding numerical rating.

Rating Scale

NUMERICAL RATING	DESCRIPTION
5	Excellent
4	Very Satisfactory
3	Satisfactory
2	Unsatisfactory
1	Needs Improvement

INDICATOR	5	4	3	2	1
A. Usability					
1. Appropriateness recognizability - degree to which users recognize the appropriateness of the website based on their needs					
2. Learnability - degree to which the website provides self-explanatory user-interface; entry fields contain default/placeholder texts; and error messages provided are understandable					

**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

DCS-SEI Form No. 2

INDICATOR	5	4	3	2	1
A. Usability					
3. Operability - degree to which the website is easy to operate, control, and can be used appropriately					
4. User error protection - the website is able to perform user entry validation and allows the user to recover from errors					
5. User interface aesthetics - degree to which the user interface enables pleasing and satisfying interaction with the user					
B. Effectiveness					
1. The website allows the users to achieve specified goals when used in specified contexts					
C. Efficiency					
1. The degree to which resources are consumed with accuracy and completeness of helping users achieve the goals					
D. Satisfaction					
1. Usefulness - degree to which the user is satisfied on perceived achievement of pragmatic goals including the results and consequences of using the website					
2. Trust - degree to which a user or other stakeholders are confident that the website will behave as it is intended					
3. Pleasure - degree to which a user gains pleasure upon fulfillment of personal needs					
4. Comfort - degree to which the user is satisfied with physical comfort					
E. Context Coverage					
1. Content completeness - degree to which the website can be used effectively, efficiently, and satisfactorily in all specified contexts of use					
2. Flexibility - degree to which the website can be used effectively, efficiently, and satisfactorily in contexts beyond those initially specified in the requirements					

Remarks:

_____

_____

_____

_____

Certified true and correct:

_____ _____

Signature over printed name of evaluator Date

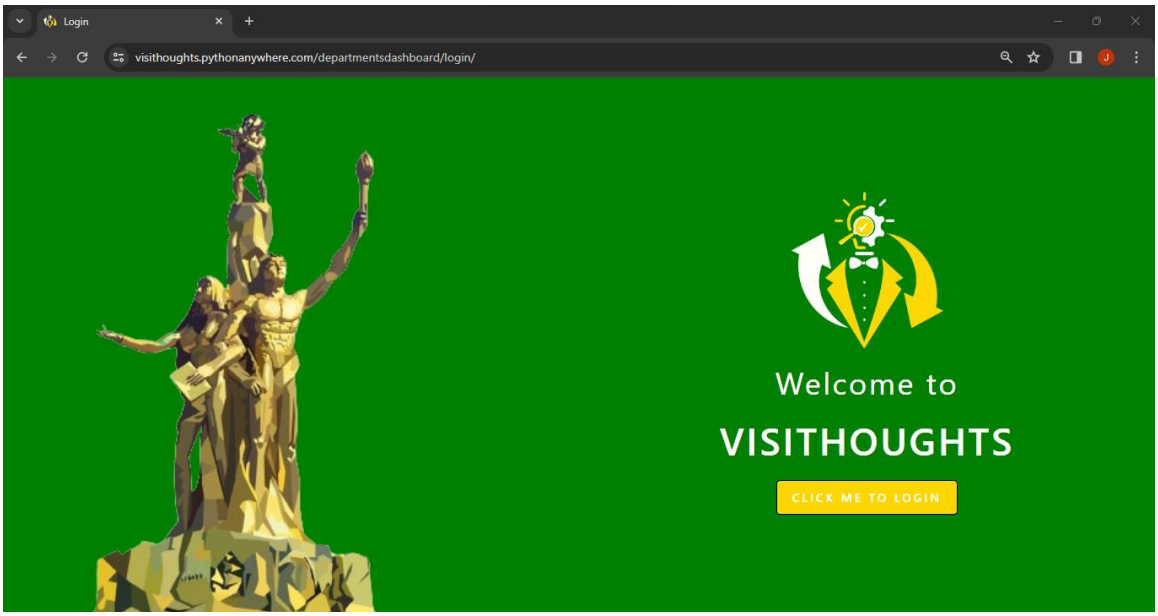
**Adapted with modification from the International Organization for Standardization and International Electro-technical Commission (ISO-IEC) 25010.*

APPENDIX K
USER'S MANUAL

Welcome to the discussion on the step-by-step procedure on how you can set up the VISITHOUGHTS: A Web Application for Customers feedback with Sentiment Analysis of CvSU-CCAT Campus.

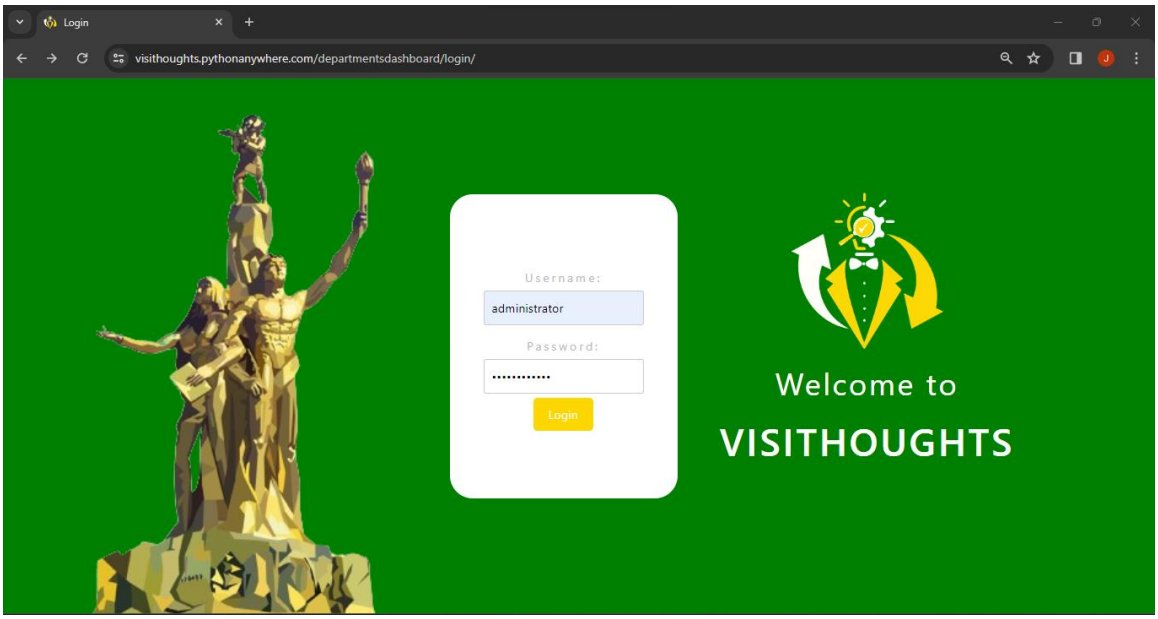
For Administration

Step 1: Open the Link for the Admin page provided by the system.



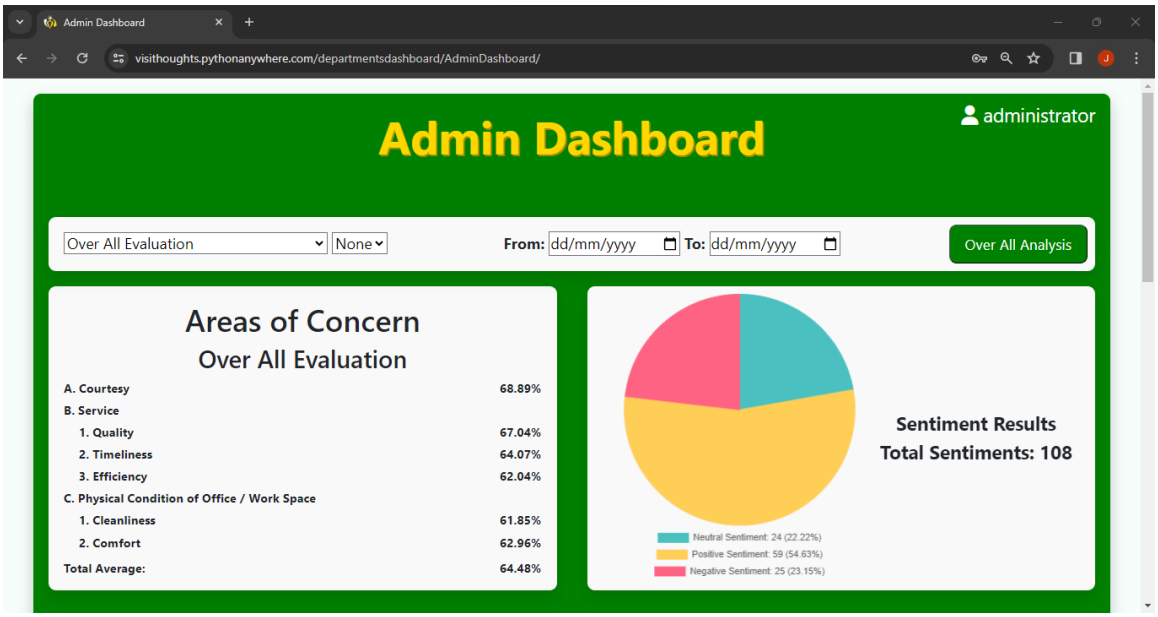
Landing page for Admin

Step 2: Input the predefined account given by the super admin.



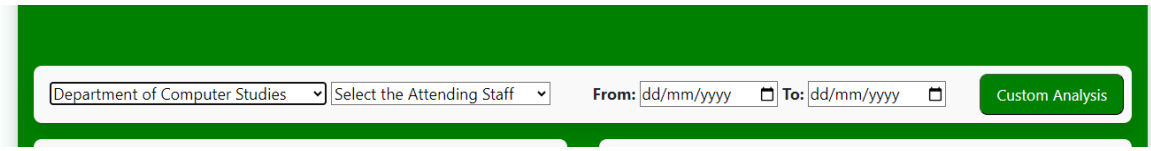
Login page

Step 3: After logging in, it will direct to admin dashboard page.



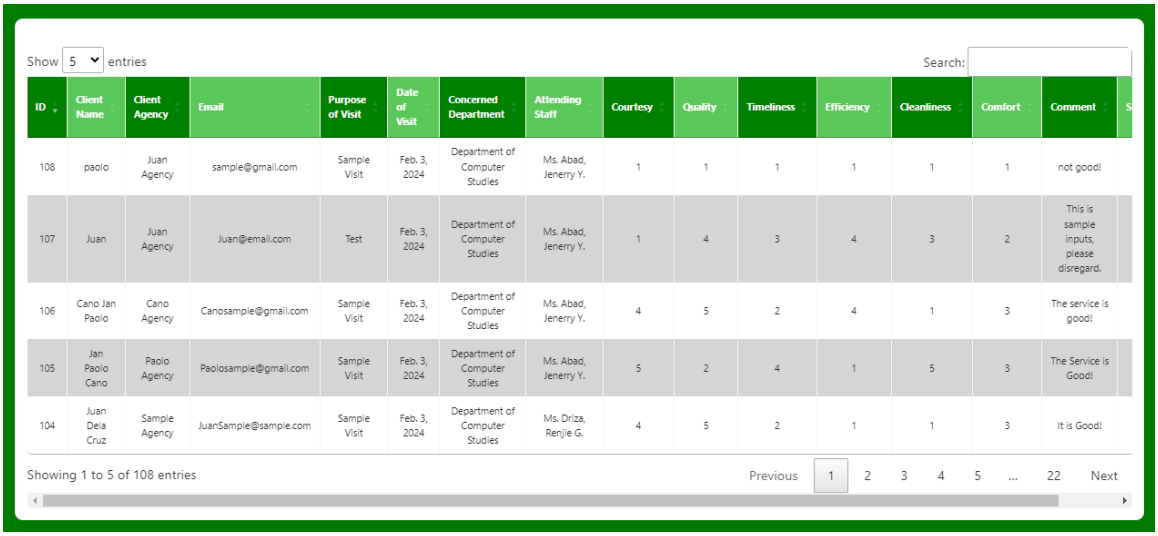
Admin dashboard page

Step 4: Administrator can choose the data that they want to filter.



Admin dashboard filtering data

Step 5: Administrator can search the data of the overall users feedbacks.



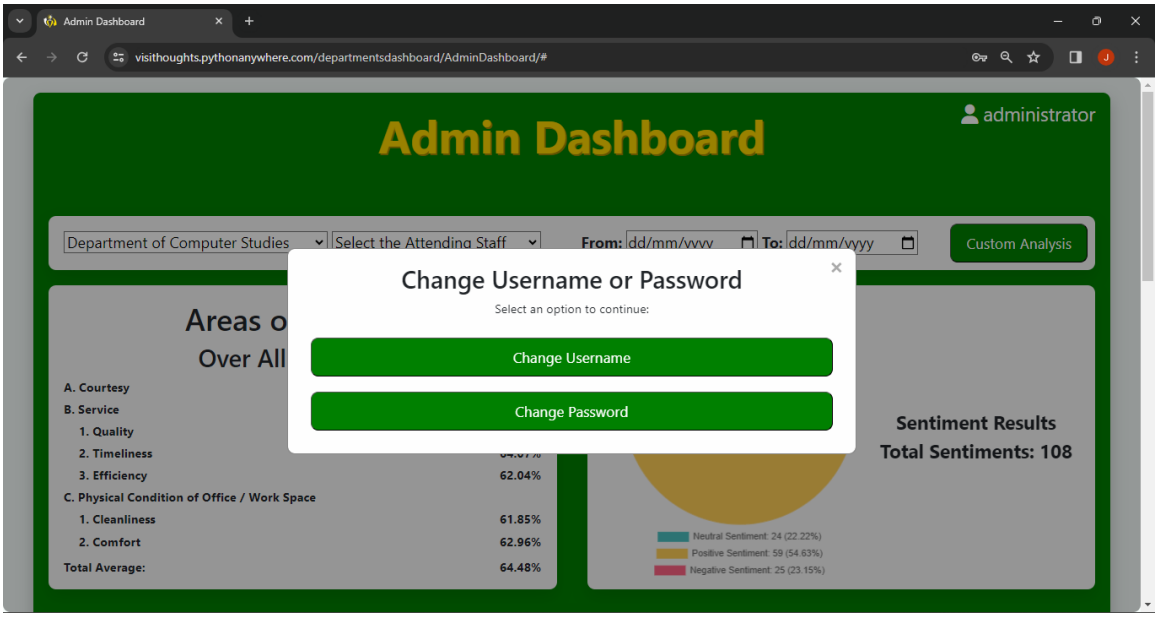
Show 5 entries Search:

ID	Client Name	Client Agency	Email	Purpose of Visit	Date of Visit	Concerned Department	Attending Staff	Courtesy	Quality	Timeliness	Efficiency	Cleanliness	Comfort	Comment
108	paolo	Juan Agency	sample@gmail.com	Sample Visit	Feb. 3, 2024	Department of Computer Studies	Ms. Abad, Jenerry Y.	1	1	1	1	1	1	not good!
107	Juan	Juan Agency	Juan@email.com	Test	Feb. 3, 2024	Department of Computer Studies	Ms. Abad, Jenerry Y.	1	4	3	4	3	2	This is sample inputs, please disregard.
106	Cano Jan Paolo	Cano Agency	Canosample@gmail.com	Sample Visit	Feb. 3, 2024	Department of Computer Studies	Ms. Abad, Jenerry Y.	4	5	2	4	1	3	The service is good!
105	Jan Paolo Cano	Paolo Agency	Paolosample@gmail.com	Sample Visit	Feb. 3, 2024	Department of Computer Studies	Ms. Abad, Jenerry Y.	5	2	4	1	5	3	The Service is Good!
104	Juan Dela Cruz	Sample Agency	JuanSample@sample.com	Sample Visit	Feb. 3, 2024	Department of Computer Studies	Ms. Driza, Renjie G.	4	5	2	1	1	3	It is Good!

Showing 1 to 5 of 108 entries Previous 1 2 3 4 5 ... 22 Next

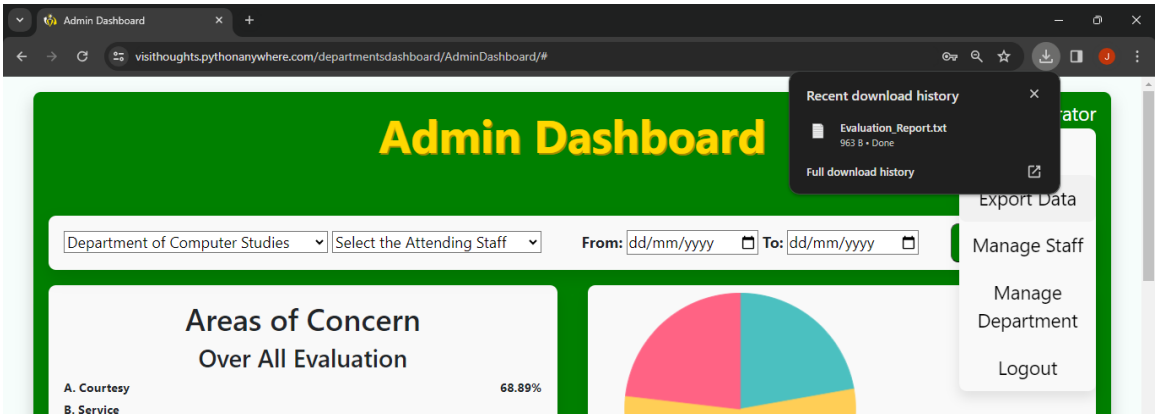
Admin Dash Board tables of user's information and feedbacks

Step 6: Administrator can change the username and password.



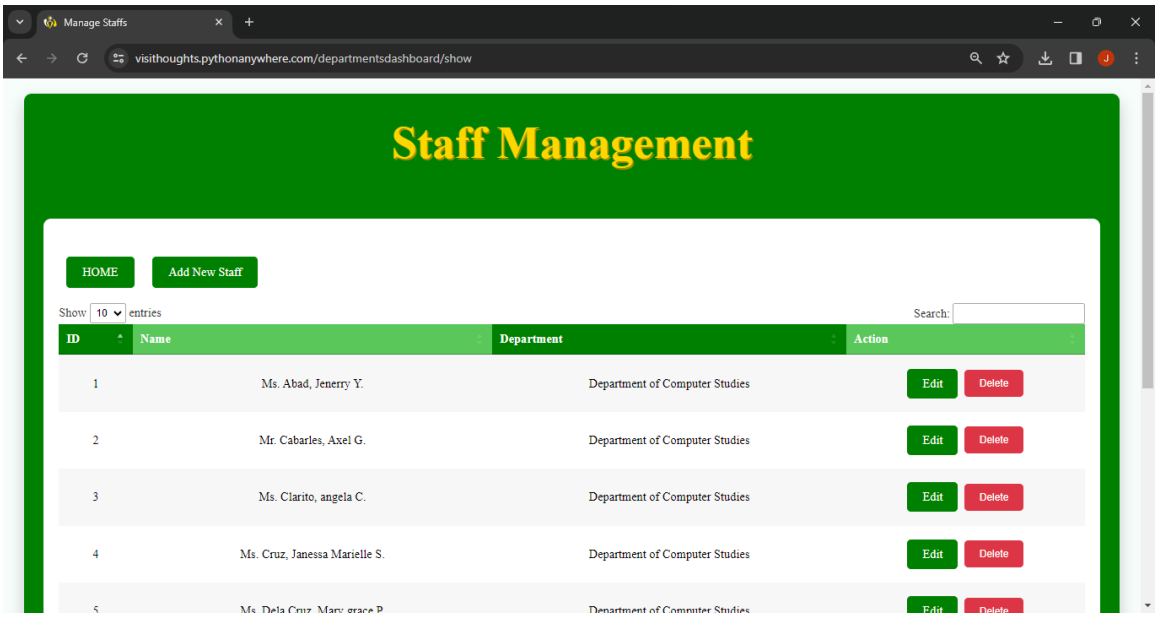
Change username and password prompt

Step 7: Administrator can export the existing statistical result.



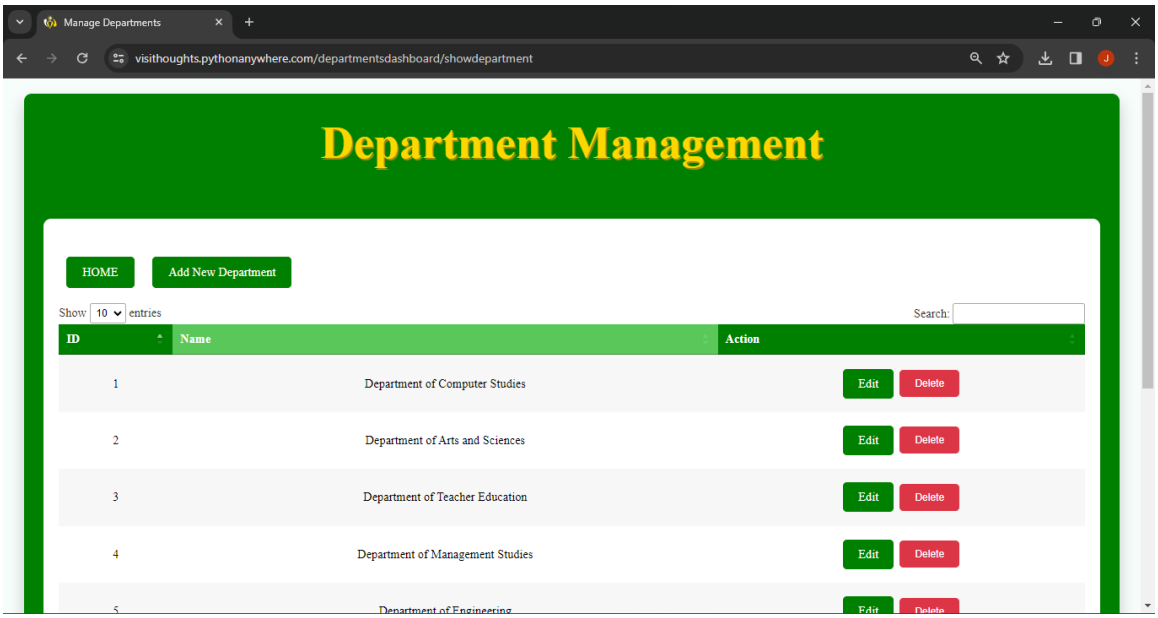
Administrator Exporting data

Step 8: Administrator can manage staff members in terms of changes.



Administrator edit, delete, add staff details

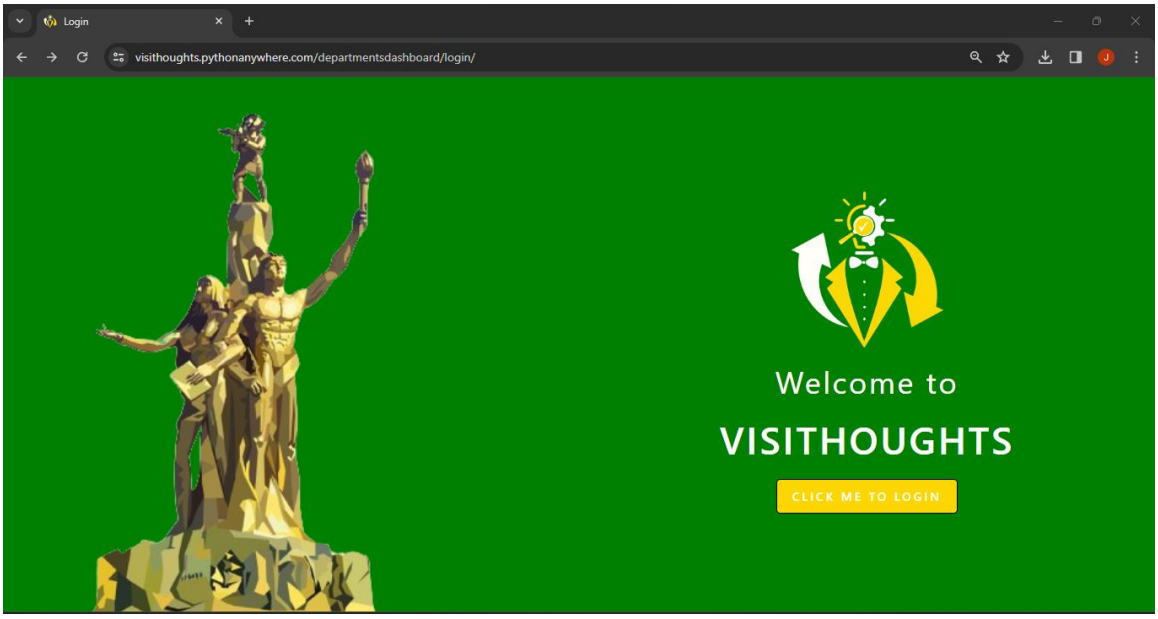
Step 9: Administrator can manage Department and Offices in terms of changes.



Administrator edit, delete, add Department and Offices details

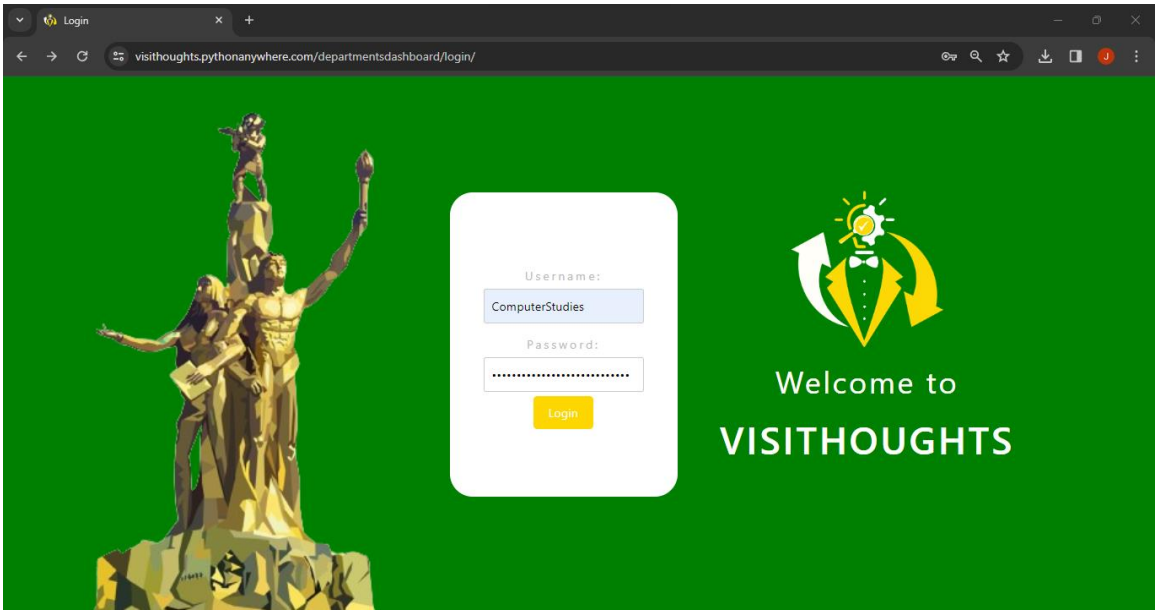
For Department Head

Step 1: Open the Link for the Department Head page provided by the system.



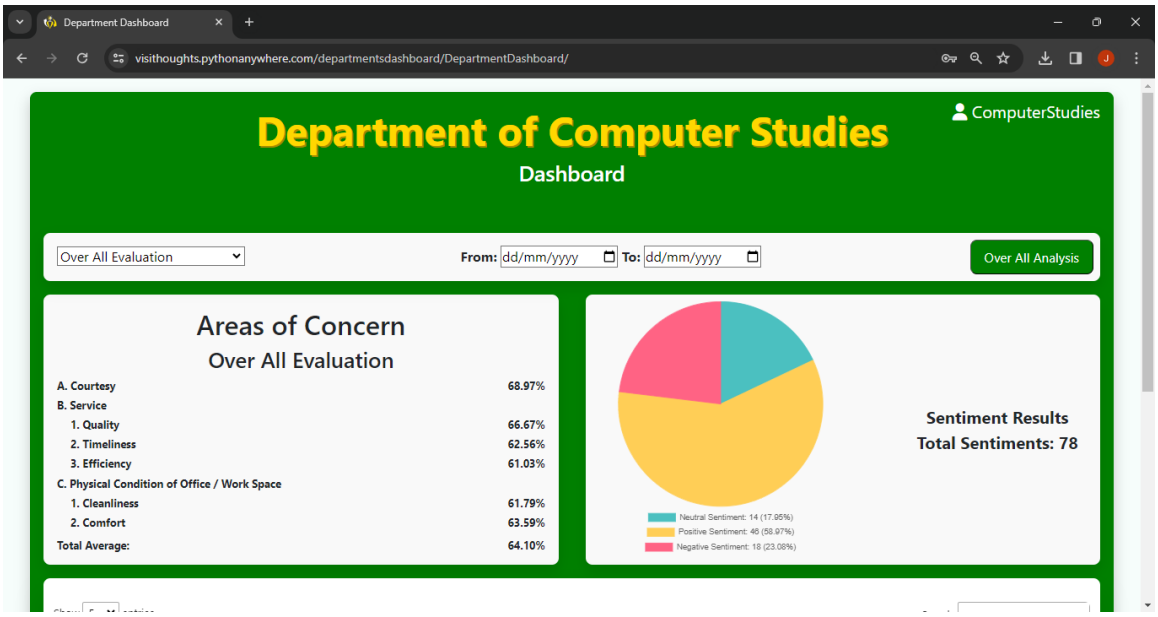
Landing page for Department Head

Step 2: Input the predefined account given by the super admin.



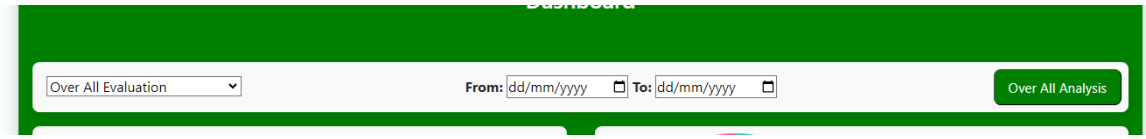
Login page

Step 3: After logging in, it will direct to Department Head dashboard page.



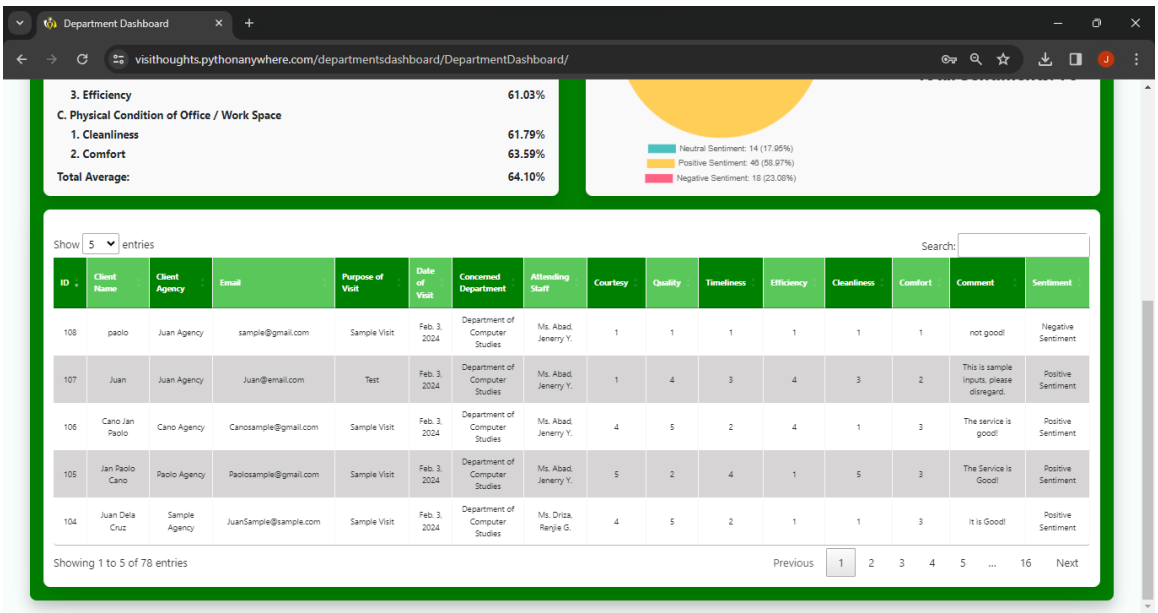
Department Head dashboard page

Step 4: Department Head can choose the data that they want to filter.



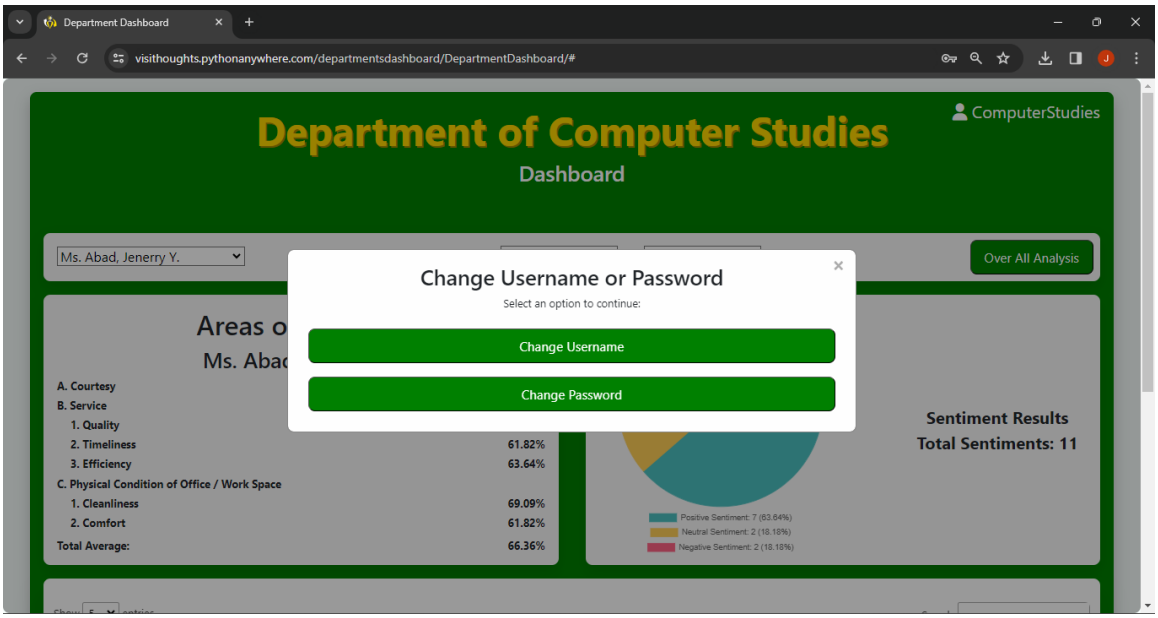
Department Head dashboard filtering data

Step 5: Department Head can search the data of the overall users feedbacks.



Department Head Dash Board tables of user's information and feedbacks

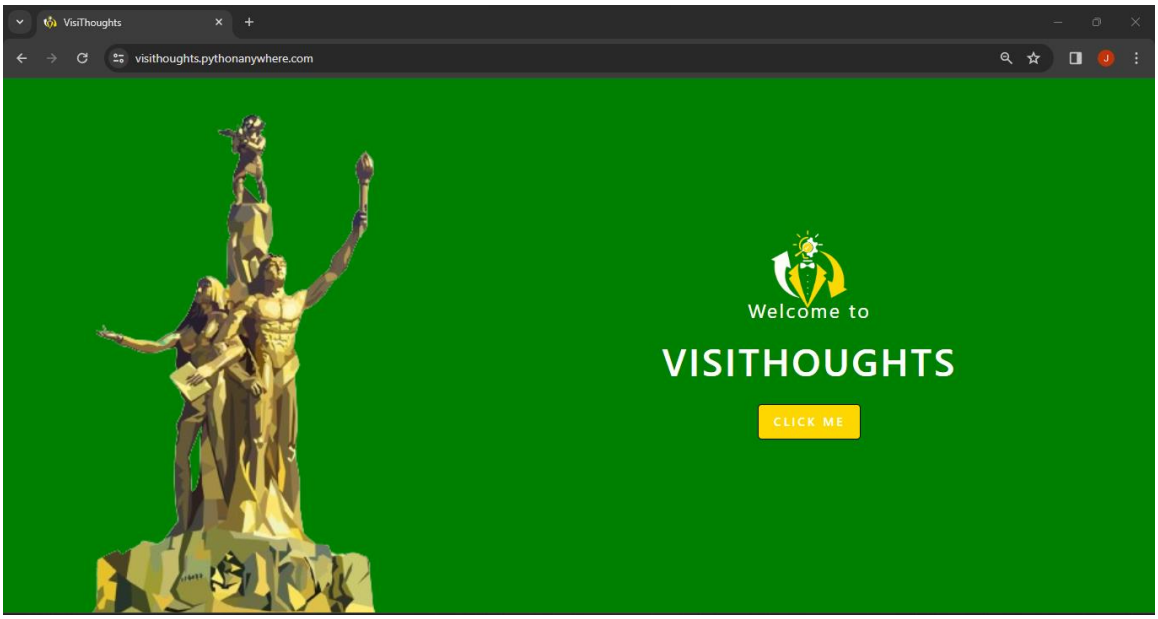
Step 6: Department Head can change the username and password.



Change username and password prompt

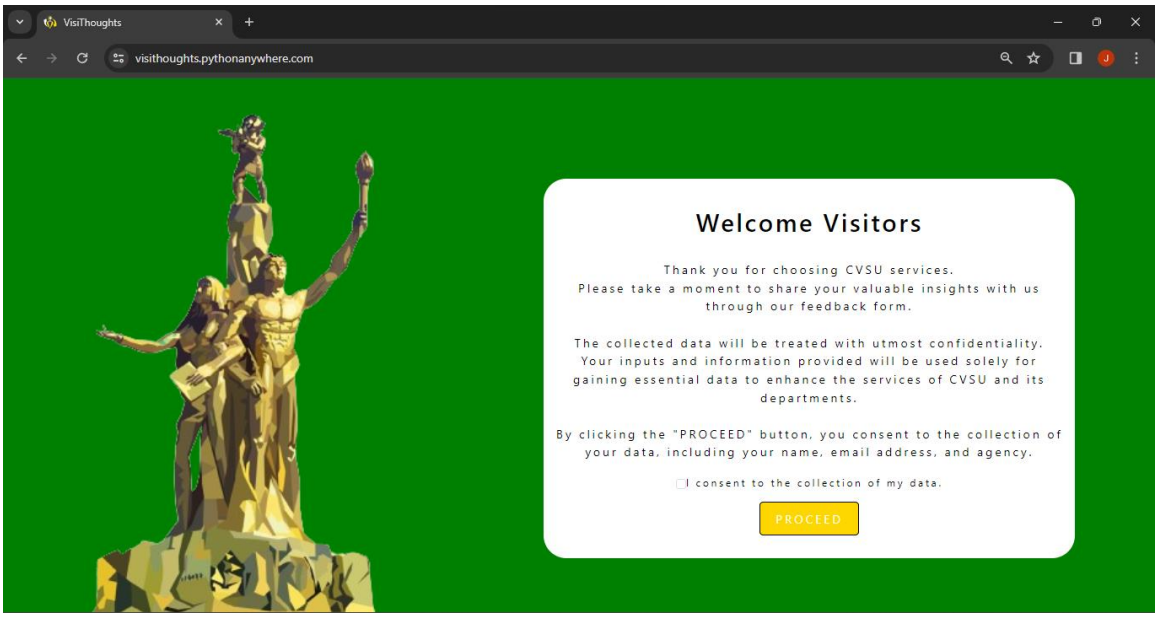
For User

Step 1: Open the Link for the Admin page provided by the system.



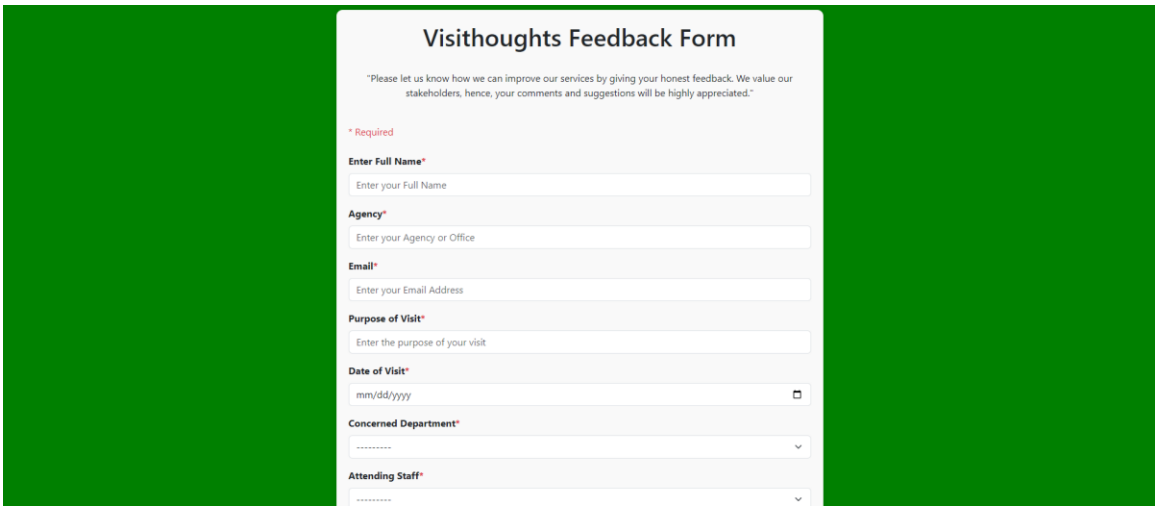
User Form Landing Page

Step 2: Users are required to check the check box before proceeding to Form.



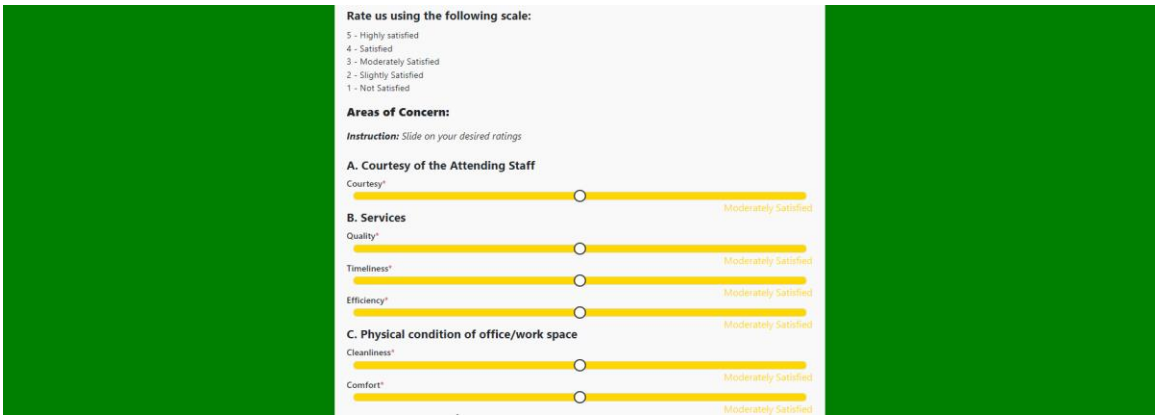
User's Form Terms and Conditions

Step 3: Users are required to fill in the information from the form.



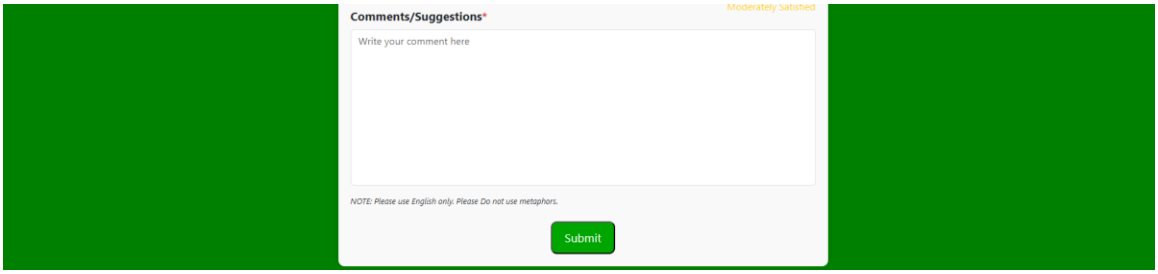
Users Form

Step 4: Users are required to slide the corresponding rates from the form.



Users Form Slider

Step 5: Users are required to write down their thoughts or feedbacks from the form.



Users Form Comments/Suggestions

APPENDIX L
CURRICULUM VITAE

ELGENE JOHN V. REYES

COMPUTER SCIENCE

As a graduate with a Bachelor's degree in Computer Science, I am eager to embark on a rewarding career in the dynamic and rapidly evolving field of technology. Armed with a solid foundation in computer science principles, programming languages, and problem-solving methodologies, I am prepared to contribute my skills and enthusiasm to a forward-thinking organization.

(+63) 993 563 7866

Tanza, Cavite

elgenereyes1515@gmail.com

https://elgenereyes1515.wixsite.com/elgenecreatives

EDUCATION

Cavite State University - CCAT (Rosario) Campus
Bachelor of Science in Computer Science
2019- 2024

St. Mary Magdalene School - Kawit
Senior High School (STEM)
With High Honor
2017-2019

Cavite State University - CCAT (Rosario) Campus
Junior High School
2013-2017

SKILLS

- Programming
- Graphic Design
- Video Editing
- Critical Thinking and Logic
- Computer Hardware

EXPERIENCES

ACHIEVEMENTS

Cavite State University – CCAT (Rosario) Campus,
President's Lister
2019-2024

Cavite State University – CCAT Research Congress
BSCS Representative
November 2023

CALABARZON Regional HackForGov Competition
Cavite State University Representative
First place
September 2023

HERU HUMAN RESOURCE SPECIALIST CO. INC.
240-hour On-the-Job Training (OJT)
July 2023-September 2023

CREOTEC Philippines Inc.
CREOTEC Immersion Center, Damarinas City, Cavite
80-hour Work Immersion
February 2019

CAREER

- Freelance Designer since 2019
- Freelance Computer Technician since 2019

LANGUAGES

F Filipino

E English

Jan Paolo Arrojado Cano

09611488879
janpaolocano01@gmail.com
Castillon Homes, Pasong Kawayan II, General Trias, Cavite

SUMMARY

Adaptable recent graduate with strong multitasking and problem-solving skills, showcasing a sense of responsibility and commitment to continuous learning. Possesses robust soft skills and is eager to contribute as a proactive team player in a challenging work environment, aiming for organizational success.

WORK EXPERIENCE

07/2023 to 09/2023

On the Job Training (OJT)
Heru Human Resource Specialist Co.

- Entered applicant data into the company's database.
- Managed LAN cables to ensure an organized and efficient network infrastructure.
- Engaged in graphic design tasks, contributing to the creation of visual materials for the company.
- Provided on-the-spot troubleshooting support for colleagues, ensuring smooth operation of internet, scanners and printers.
- Conducted quality assurance checks on data input to minimize errors and maintain data integrity.

ABOUT

BIRTHDAY: 01/16/2001
AGE: 23
BIRTHPLACE: Tondo, Manila
CITIZENSHIP: Filipino
RELIGION: Roman Catholic
HEIGHT: 169 cm
WEIGHT: 60 kg
STATUS: Single
SEX: Male

EDUCATION

David P. Jimenez Elementary School
Wawa III, Rosario Cavite
2007-2013

Bagbag National High School - Junior High
Bagbag II, Rosario Cavite
2013-2017

Rosario Institute - Senior High
C. Abueg St, Rosario, Cavite
2017-2019

CvSU- CCAT Campus - College
Bachelor of Science in Computer Science
CCAT, Rosario Cavite
2019-2024

SKILLS

- Flexible
- Multitasking
- Problem Solving Skills
- IT Tools
- Soft Skills
- Continuous Learning
- Communication

CHARACTER REFERENCE

Melvin M. Vergara
Customer Service Representative
0918-250-6793

Erika C. Belanio
Marketing Staff
0969-106-6341

I do hereby declare that all the information given above is true to the best of my knowledge and belief.

JAN PAOLO ARROJADO CANO

Karl Dohneyzel Dela Cruz

09929012254
krldlcrz1418@gmail.com
204 Manggahan St. Brgy. Niog 1 Bacoor City of Cavite

CUSTOMER SERVICE REPRESENTATIVE

To consistently achieve a customer satisfaction rating of 95% or higher by promptly addressing customer inquiries, resolving issues within a maximum of 24 hours, and fostering positive interactions. Utilize excellent communication and problem-solving skills to exceed customer expectations, contributing to the overall success of the company.

KEY COMPETENCIES

Process improvement
Data-driven strategic planning
People Management
Report writing and presenting
Critical thinking skills
Excellent communication skills
Investigation Report
Administration Task

PROFESSIONAL EXPERIENCE

CXI Services Inc.
Crew accomodation Agent/ Team Leader/ Service Level Technician **August 2021 - Present**

Facilitated customer support by coordinating hotel and transportation accommodations through both email and telephone channels. Progressed to the position of Subject Matter Expert (SME) and subsequently promoted to Team Leader, where responsibilities included administrative tasks, payroll input, agent assistance, report creation for investigations, coaching, and additional administrative duties. Interned as a Service Level Technician, overseeing roll call for employees, troubleshooting computer and company tools, and ensuring internet maintainability.

EDUCATIONAL BACKGROUND

Bachelor of Science in Computer Science
Cavite State University - CCAT Campus
2019-2024

TVL (Technical-Vocational-Livelihood) Strand
City of Bacoor Senior High School - Dulong Bayan
2017-2019
With Honor

Junior High School
City of Bacoor Junior High School - Tabing - Dagat
2013-2017

EXTRACURRICULAR ACTIVITIES

San Lorenzo Ruiz Band Inc. 2003
Musician
2016 - Present

Future Information Communication Technologist (FICT)
Public Information Officer
2022-2023

ABOUT

Age: 23
Sex: Male
Birthday: November 18, 2000
Birthplace: Bacoor City, Cavite
Height: 155 Cm
Weight: 55 Kg
Citizenship: Filipino

STATUTORY

SSS: 3499901513
TIN: 384120604
HDMF: 121246483186
PHILHEALTH: 080267528458

CHARACTER REFERENCE

Abby Banta
Civil Engineer
09158767430

Benedict Mendoza
Operations Manager
09052390379

Karl Dohneyzel Dela Cruz
I do hereby declare that all the information given above
is true to the best of my knowledge and belief.

APPENDIX M
SOURCE CODE

CLIENT SIDE

LANDING PAGE

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

<title>VisiThoughts</title>

   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>
   <link rel="stylesheet" href="{% static 'CSS/landing.css' %}">
   <!-- favicon <dev elgene -->
   <link rel="icon" href="{% static 'elgene.png' %}" type="image/x-icon">
   <link rel="shortcut icon" href="{% static 'favicon.ico' %}" type="image/x-icon">
</head>

<body>
   <img src="{% static 'layadiwa.png' %}" alt="Background Image" class="background-image">
   <img src="{% static 'elgene.png' %}" alt="Logo" class="logo">
   <div class="wrapper">
      <span class="button-text">
         <p>Welcome to</p>

<h1>VisiThoughts</h1>
      </span>
      <div class="button" onclick="document.body.classList.add('active')">
         <span class="button-style2">
            <p>Click Me</p>
         </span>

         <!-- <div class="button-backgrounds">
            <div class="button-circle button-circle1"></div>
            <div class="button-circle button-circle2"></div>
            <div class="button-circle button-circle3"></div>
            <div class="button-circle button-circle4"></div>
         </div> -->
      </div>
   </div>

   <div class="wrapper">
      <div class="popup">
         <div class="popup-inside">
            <div class="backgrounds">
               <div class="background"></div>
               <div class="background background2"></div>
               <div class="background background3"></div>
```

```
                <div
class="background
background4"></div>
                <div
class="background
background5"></div>
                <div
class="background
background6"></div>
            </div>
        </div>
        <div
class="content">
            <div
class="content-
wrapper">

<h2>Welcome
Visitors</h2>
                <p
class="second-part-
paragraph">

<br>Thank you for
choosing CVSU
services.

<br>Please take a
moment to share your
valuable insights with us
through our feedback
form.
                <br>
                <br>The
collected data will be
treated with utmost
confidentiality.

<br>Your inputs and
information provided
will be used solely for
gaining essential data to
enhance the services of
CVSU and its
departments.
                <br>
                <br>By
clicking the
"PROCEED" button,
you consent to the
collection of your data,
including your name,
email address, and
agency.
                </p>
                <!--
Checkbox added here -->
                <div
class="form-check d-
flex justify-content-
center align-items-
center" style="font-size:
smaller; margin-bottom:
25px;">
                <input
class="form-check-
input" type="checkbox"
id="consentCheckbox"
required>
                <label
class="form-check-
label"
for="consentCheckbox"
>
                    I
consent to the collection
of my data.
                </label>
            </div>
            <a
href="{% url 'add' %}"
id="proceedButton"
class="button-style"
disabled>PROCEED</a
>
            </div>
        </div>
    </div>
</div>
<script>
  // Get the checkbox
and "PROCEED" button
  var checkbox =
document.getElementBy
Id('consentCheckbox');
  var proceedButton =
document.getElementBy
Id('proceedButton');

  // Add an event
listener to the checkbox

checkbox.addEventListe
ner('change', function ()
{
    // Enable/disable
the "PROCEED" button
based on checkbox state

proceedButton.disabled
= !checkbox.checked;
  });

  // Add an event
listener to the
"PROCEED" button to
check if the checkbox is
checked before
proceeding

proceedButton.addEvent
Listener('click', function
(event) {
    if
(!checkbox.checked) {

event.preventDefault();
// Prevent the default
click action if the
checkbox is not checked
        alert('Please
consent to the collection
of your data before
proceeding.');
    }
  });
</script>
```

```
</body>
</html>
```

FORM

```
{% load static %}
<!doctype html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>
   <link rel="stylesheet" href="{% static 'CSS/Form_style.css' %}">
   <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

   <title>Visithoughts Feedback Form</title>
   <!-- favicon <develgene -->
   <link rel="icon" href="{% static 'elgene.png' %}" type="image/x-icon">
   <link rel="shortcut icon" href="{% static 'favicon.ico' %}" type="image/x-icon">
</head>
<body>
<div class="container mt-4">

   <form method="post" id="FeedbackForm" data-staffs-url="{% url 'ajax_load_staffs' %}" class="form-horizontal" role="form">
      {% csrf_token %}

      <center>

<h1>Visithoughts Feedback Form</h1>
         <br>
         <p>
            <italic>
            "Please let us know how we can improve our services by giving your honest feedback. We value our stakeholders, hence,
            your comments and suggestions will be highly appreciated."
            </italic>
         </p>
      </center>
      <br>
      <span class="text-danger">* Required</span>
      <br>
      <br>

      <div class="mb-3">
         <label for="fullName" class="form-label">Enter Full Name<span class="text-danger">*</span></label>
         {{ form.name }}
      </div>

      <div class="mb-3">
         <label for="agency" class="form-label">Agency<span class="text-danger">*</span></label>
         {{ form.agency }}
      </div>

      <div class="mb-3">
         <label for="email" class="form-
```

```
label">Email<span
class="text-
danger">*</span></labe
l>
        {{ form.email }}
    </div>

    <div class="mb-
3">
        <label
for="purposeOfVisit"
class="form-
label">Purpose of
Visit<span class="text-
danger">*</span></labe
l>
        {{
form.purposeOfVisit }}
    </div>

    <div class="mb-
3">
        <label
for="dateOfVisit"
class="form-label">Date
of Visit<span
class="text-
danger">*</span></labe
l>
        {{
form.dateOfVisit }}
    </div>

    <div class="mb-
3">
        <label
for="concernedDepartm
ent" class="form-
label">Concerned
Department<span
class="text-
danger">*</span></labe
l>
        {{
form.department }}
    </div>

    <div class="mb-
3">
        <label
for="attendingStaff"
class="form-
label">Attending
Staff<span class="text-
danger">*</span></labe
l>
        {{ form.staff }}
    </div>

    <div class="mb-
3">
        <h1
class="subtext">Rate us
using the following
scale:</h1>
        <p
class="paragraph">
            5 - Highly
satisfied<br>
            4 -
Satisfied<br>
            3 - Moderately
Satisfied<br>
            2 - Slightly
Satisfied<br>
            1 - Not
Satisfied
        </p>
    </div>

    <div class="mb-
3">
        <h1
class="subtext1"><stron
g>Areas of
Concern:</strong></h1>
    </div>

    <div>

<i><b>Instruction:</b>
Slide on your desired
ratings<br><br></i>
    </div>

    <div class="mb-
3">
        <h2
class="subtext1">A.
Courtesy of the
Attending Staff</h2>
        <h2
class="rating-
text">Courtesy<span
class="text-
danger">*</span></h2>
        <div
class="rating">
            {{
form.courtesy }}
        </div>
        <div
class="satisfactionText"
style="float:
right;"></div>
    </div>

    <div class="mb-
3">
        <h2
class="subtext1">B.
Services</h2>
        <h2
class="rating-
text">Quality<span
class="text-
danger">*</span></h2>
        <div
class="rating">
            {{
form.quality }}
        </div>
        <div
class="satisfactionText2
"  style="float:
right;"></div>
    </div>
```

```
        <div class="mb-3">
            <h2 class="rating-text">Timeliness<span class="text-danger">*</span></h2>
            <div class="rating">
                {{ form.timeless }}
            </div>
            <div class="satisfactionText3" style="float: right;"></div>
        </div>

        <div class="mb-3">
            <h2 class="rating-text">Efficiency<span class="text-danger">*</span></h2>
            <div class="rating">
                {{ form.efficiency }}
            </div>
            <div class="satisfactionText4" style="float: right;"></div>
        </div>

        <div class="mb-3">
            <h2 class="subtext1">C. Physical condition of office/work space</h2>
            <h2 class="rating-text">Cleanliness<span class="text-danger">*</span></h2>
            <div class="rating">
                {{ form.cleanliness }}
            </div>
            <div class="satisfactionText5" style="float: right;"></div>
        </div>

        <div class="mb-3">
            <h2 class="rating-text">Comfort<span class="text-danger">*</span></h2>
            <div class="rating">
                {{ form.comfort }}
            </div>
            <div class="satisfactionText6" style="float: right;"></div>
        </div>

        <div class="mb-3">
            <h2 class="subtext1">Comments/Suggestions<span class="text-danger">*</span></h2>
            <div class="comment-box">
                {{ form.comment }}
            </div>
            <p style="font-size: 13px; font-style: italic;">NOTE: Please use English only. Please Do not use metaphors.</p>
        </div>

        <div class="d-flex justify-content-center mt-4">
            <input type="submit" class="btn-custom" value="Submit">
        </div>
    </form>

</div>
<script>

$("#id_department").change(function () {
    const url = $("#FeedbackForm").attr("data-staffs-url");  // get the url of the `load_staffs` view
    const departmentId = $(this).val();  // get the selected country ID from the HTML input

// <deve elgene>
    $.ajax({  // initialize an AJAX request
        url: url,  // set the url of the request (= /feedback/ajax/load-staffs/ )
        data: {

'department_id': departmentId  // GET parameters
        },
```

```
            success: function
(data) {

$("#id_staff").html(data)
;
            }
        });

    });


$(document).ready(funct
ion() {
        var rating1Slider =
$('#Rating1');
        var rating2Slider =
$('#Rating2');
        var rating3Slider =
$('#Rating3');
        var rating4Slider =
$('#Rating4');
        var rating5Slider =
$('#Rating5');
        var rating6Slider =
$('#Rating6');

        // Set the default
colors and text on load

setDefaultSliderAppeara
nce();


rating1Slider.on('input',
function() {

updateSliderAppearance
($(this));
        });

rating2Slider.on('input',
function() {

updateSliderAppearance
($(this));
        });

rating3Slider.on('input',
function() {

updateSliderAppearance
($(this));
        });

rating4Slider.on('input',
function() {

updateSliderAppearance
($(this));
        });

rating5Slider.on('input',
function() {

updateSliderAppearance
($(this));
        });

rating6Slider.on('input',
function() {

updateSliderAppearance
($(this));
        });

        function
setDefaultSliderAppeara
nce() {
            // Call
updateSliderAppearance
for each slider to set the
default text

updateSliderAppearance
(rating1Slider);

updateSliderAppearance
(rating2Slider);

updateSliderAppearance
(rating3Slider);

updateSliderAppearance
(rating4Slider);

updateSliderAppearance
(rating5Slider);

updateSliderAppearance
(rating6Slider);
        }
    });

    function
updateSliderAppearance
(slider) {
        var value =
slider.val();
        var color =
getColorForValue(value
);
        var text =
getValueText(value);


slider.css('background-
color', color);

slider.siblings('.rating-
text').text(text);
    }

    function
getValueText(value) {
        switch (value) {
            case '1':
                return 'Not
Satisfied';
            case '2':
                return 'Slightly
Satisfied';
            case '3':
                return
'Moderately Satisfied';
            case '4':
                return
'Satisfied';
```

```
        case '5':
            return 'Highly Satisfied';
        default:
            return '';
    }
}

function getColorForValue(value) {
    switch (value) {
        case '1':
            return 'red';
        case '2':
            return 'orange';
        case '3':
            return '#FFD700';
        case '4':
            return '#76BA1B';
        case '5':
            return 'green';
        default:
            return 'transparent'; // Handle other values if needed
    }
}
</script>
<script src="{% static 'JS\Form_script.js' %}"></script>
</body>
</html>
```

DROP DOWN LIST

```
<option value="" disabled selected>Select the Attending Staff</option>
{% for staff in staffs %}
<option value="{{ staff.pk }}">{{ staff.name }}</option>
{% endfor %}
```

AFTER FORM SUBMISSION

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

    <title>Feedback Submitted</title>
    <link rel="stylesheet" href="{% static 'CSS/formsubmitted.css' %}">
    <!-- favicon <dev elgene -->
    <link rel="icon" href="{% static 'elgene.png' %}" type="image/x-icon">
    <!-- <link rel="shortcut icon" href="{% static 'favicon.ico' %}" type="image/x-icon"> -->
</head>
<body>
    <div class="wrapper">
        <h1>Thank You!</h1>
        <p>
            <br>Your input means the world to us. We appreciate you taking the time
            <br> to share your thoughts and suggestions.
            <br>Your feedback helps us grow, learn, and better serve our community.
            <br>
            <br>At VisiThoughts, we're committed to making improvements based on your feedback,
            <br>and your voice matters.If you have any more ideas or thoughts in the future,
            <br>please don't hesitate to reach out.
```

```
We're here to listen and act on your feedback.
        <br>
        <br>Thank you for being a valued member of our community.
      </p>
    </div>
</body>
</html>
```

ADMIN.PY

```
from django.contrib import admin

# Register your models here.
from visithoughtsForm.models import Staff, Department, Feedback

admin.site.register(Staff)
admin.site.register(Department)
admin.site.register(Feedback)
```

FORMS.PY

```
from django import forms

from visithoughtsForm.models import Feedback, Staff, Department

class FeedbackCreationForm(forms.ModelForm):
    name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter your Full Name'}),
    )
    agency = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter your Agency or Office'}),
    )
    email = forms.EmailField(
        widget=forms.EmailInput(attrs={'class': 'form-control', 'placeholder': 'Enter your Email Address'}),
    )
    purposeOfVisit = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter the purpose of your visit'}),
    )
    dateOfVisit = forms.DateField(
        widget=forms.DateInput(attrs={'class': 'form-control', 'type':'date'}),
    )
    department = forms.ModelChoiceField(
        queryset=Department.objects.all(),
        label='Concerned Department',
        widget=forms.Select(attrs={'class': 'form-select'}),
    )
    staff = forms.ModelChoiceField(
        queryset=Staff.objects.all(),
        label='Attending Staff',
        widget=forms.Select(attrs={'class': 'form-select'}),
    )
    courtesy = forms.IntegerField(
        widget=forms.NumberInput(attrs={'class': 'form-control', 'type': 'range', 'min': '1', 'max': '5', 'value':'3', 'id':'Rating1'}),
    )
    quality = forms.IntegerField(
        widget=forms.NumberInput(attrs={'class': 'form-control', 'type': 'range', 'min': '1', 'max': '5', 'value':'3', 'id':'Rating2'}),
    )
    timeless = forms.IntegerField(
        widget=forms.NumberInput(attrs={'class': 'form-control', 'type': 'range',
```

```
'min': '1', 'max': '5', 'value':'3', 'id':'Rating3'}),
    )
    efficiency = forms.IntegerField(

widget=forms.NumberInput(attrs={'class': 'form-control', 'type': 'range', 'min': '1', 'max': '5', 'value':'3', 'id':'Rating4'}),
    )
    cleanliness = forms.IntegerField(

widget=forms.NumberInput(attrs={'class': 'form-control', 'type': 'range', 'min': '1', 'max': '5', 'value':'3', 'id':'Rating5'}),
    )
    comfort = forms.IntegerField(

widget=forms.NumberInput(attrs={'class': 'form-control', 'type': 'range', 'min': '1', 'max': '5', 'value':'3', 'id':'Rating6'}),
    )
    comment = forms.CharField(

widget=forms.Textarea(attrs={'class': 'form-control', 'placeholder': 'Write your comment here',}),
    )

    class Meta:
        model = Feedback
        fields = '__all__'


    def __init__(self, *args, **kwargs):

super().__init__(*args, **kwargs)

self.fields['staff'].queryset = Staff.objects.none()

        if 'department' in self.data:
            try:
                department_id = int(self.data.get('department'))

self.fields['staff'].queryset = Staff.objects.filter(department_id=department_id).order_by('name')
            except (ValueError, TypeError):
                pass  # invalid input from the client; ignore and fallback to empty staff queryset <dev elgene>
        elif self.instance.pk:

self.fields['staff'].queryset = self.instance.department.staff_set.order_by('name')
```

MODELS.PY

```
from django.db import models

#structure of department table <dev elgene>
class Department(models.Model):
    name = models.CharField(max_length=40)

    def __str__(self):
        return self.name

    class Meta:
        db_table = 'visithoughtsform_department'

#structure of staff table <dev elgene>
class Staff(models.Model):
    department = models.ForeignKey(Department, on_delete=models.CASCADE)
    name = models.CharField(max_length=40)

    def __str__(self):
        return self.name

    class Meta:
        db_table = 'visithoughtsform_staff'

#structure of feedback table <dev elgene>
class Feedback(models.Model):
```

```
    name = models.CharField(max_length=124)
    agency = models.CharField(max_length=100)
    email = models.CharField(max_length=125)
    purposeOfVisit = models.CharField(max_length=125)
    dateOfVisit = models.DateField()
    department = models.CharField(max_length=40, blank=True, null=True)
    staff = models.CharField(max_length=40, blank=True, null=True)

courtesy=models.PositiveIntegerField()

quality=models.PositiveIntegerField()

timeless=models.PositiveIntegerField()

efficiency=models.PositiveIntegerField()

cleanliness=models.PositiveIntegerField()

comfort=models.PositiveIntegerField()

comment=models.TextField()

sentiment=models.CharField(max_length=20, null=True, blank=True)

    class Meta:
        db_table = 'visithoughtsform_feedback'
```

URLS.PY

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.landing, name='landing'),
    path('form/', views.create_view, name='add'),
    path('formsubmitted/', views.formsubmitted, name='formsubmitted'),

path('feedback/ajax/load-staffs/', views.load_staffs, name='ajax_load_staffs'), # AJAX
]
```

VIEWS.PY

```
from django.shortcuts import render, redirect
from .forms import FeedbackCreationForm
from .models import Staff

#landing
def landing(request):
    return render(request, 'landing.html')
#landing

# -- Your view for creating Feedback objects -- <dev elgene>
def create_view(request):
    form = FeedbackCreationForm()
    if request.method == 'POST':
        form = FeedbackCreationForm(request.POST)
        if form.is_valid():
            # Get the selected department and staff objects
            department = form.cleaned_data['department']
            staff = form.cleaned_data['staff']

            # Extract the names of the department and staff

department_name = department.name
            staff_name = staff.name

            # Extract the comment text from the form
            comment_text = form.cleaned_data['comment']

            # Perform sentiment analysis on the comment text
            sentiment = perform_sentiment_analysis(comment_text)
```

```
print("Sentiment:", sentiment)  # Add this debug print

        # Save the record with department and staff names and sentiment
        feedback = form.save(commit=False)  # Don't save it to the database yet

feedback.department = department_name
        feedback.staff = staff_name

feedback.sentiment = sentiment  # Assign the sentiment value
        feedback.save()  # Now save it to the database

        return redirect('formsubmitted')
    return render(request, 'home.html', {'form': form})

#--->AJAX <dev elgene>
def load_staffs(request):
    department_id = request.GET.get('department_id')
    staffs = Staff.objects.filter(department_id=department_id).all()
    return render(request, 'staff_dropdown_list_options.html', {'staffs': staffs})

#--->end of AJAX

#--->start of sentiment Analysis <dev elgene>
import string
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.sentiment.vader import SentimentIntensityAnalyzer

def perform_sentiment_analysis(text):
    # Convert the text to lower case
    elgene = text.lower()

    # Remove punctuation from the text.
    reyes = elgene.translate(str.maketrans('', '', string.punctuation))

    # Tokenization of the words using the nltk library
    tokenized_words = word_tokenize(reyes, "english")

    # Remove words that do not add meaning to the sentiment
    final_words = [word for word in tokenized_words if word not in stopwords.words('english')]

    # Perform sentiment analysis using VADER
    score = SentimentIntensityAnalyzer().polarity_scores(reyes)
    neg = score['neg']
    pos = score['pos']

    if neg > pos:
        return "Negative Sentiment"
    elif pos > neg:
        return "Positive Sentiment"
    else:
        return "Neutral Sentiment"
#--->end of sentiment analysis

#--->after submission <dev elgene>
def formsubmitted(request):
    # Your view logic here
    return render(request, 'formsubmitted.html')
#--->end after submission
```

FORM STYLE CSS

```
    body{
        background-color: green;
    }
    .container {
        max-width: 800px;
        margin: auto;
        padding: 20px;
        background-color: #f9f9f9;
        border-radius: 10px;
```

```
        box-shadow:    0    0
10px rgba(0, 0, 0, 0.1);
        margin-bottom:
20px;
    }
    .comment-box
textarea {
        resize: none;
    }
    .text-danger {
        color: red;
    }
    .form-label {
        font-weight: bold;
    }

    .form-control {
        margin-bottom:
15px;
    }

    .form-select {
        margin-bottom:
15px;
    }

    .rating {
        display: flex;
        align-items: center;
    }

    .rating input {
        flex-grow: 1;
        margin: 0 10px;
    }

    /*    Custom    Button
Style */
    .btn-custom {
        background-color:
rgb(0, 165, 0);
        color: #fff;
        font-size: 1.2em;
        padding:          10px
20px;
        border-radius:
10px;
        box-shadow: none;
    }
    .btn-custom:hover {
        background-color:
rgb(90, 199, 90);
        color: #fff;
        font-size: 1.2em;
        padding:          10px
20px;
        border-radius:
10px;
        box-shadow: none;
    }

    /* Additional Custom
Styles */
    .subtext, .subtext1 {
        font-size: 1.2em;
        font-weight: bold;
        margin-bottom:
10px;
    }

    .paragraph {
        font-size: 0.9em;
    }

    .line {
        border-top:       1px
solid #ccc;
        margin: 20px 0;
    }
    .rating-text {
        margin-bottom:
5px;
    }
    .rating-text {
        margin-bottom:
5px;
        font-size: 0.9em;
    }


#Rating1,#Rating2,#Rati
ng3,#Rating4,#Rating5,
#Rating6 {

        width: 100%;
        height: 10px;

    }
    /* Style the thumb */
    input[type="range"]::-
webkit-slider-thumb {
        -webkit-
appearance: none;
        appearance: none;
        width: 20px;
        height: 20px;
        background: #fff;
        border:   2px   solid
#333;
        border-radius: 50%;
        cursor: pointer;
        transition:
background          0.2s,
transform 0.2s;
    }
    /* Hover effect for the
thumb */
    input[type="range"]::-
webkit-slider-
thumb:hover {
        border:   2px   solid
#fff;
        background: #333;
        transform:
scale(1.2);  /*  Make  the
thumb slightly bigger on
hover */
    }
    .comment-input {
        width: 100%;
        height: 100px;
    }

    /*  Responsive  Styles
*/
    @media  (max-width:
768px) {
```

```
        .container {
            max-width:
100%;
        }
    }

    @media (max-width:
576px) {
        .container {
            margin-left: auto;
            margin-right:
auto;
        }
        .rating-text,
.subtext, .subtext1,
.paragraph {
            font-size: 1em;
        }
        .comment-input {
            height: 150px;
        }
    }
```

AFTER FORM SUBMISSION CSS

```
* {
    box-sizing: border-
box;
}

body {
    background: green;
    font-family: 'Arial',
sans-serif;
    user-select: none;
    letter-spacing: 1px;
    color: #000000;
    margin: 0;
}

.color {
    background: linear-
gradient(to right,
#e1e0ff, #d7fadd);
}

.wrapper {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    border-radius: 10px;
    background-color:
#ffffff;
    box-shadow: 0 0 20px
rgba(0, 0, 0, 0.2);
    text-align: center;
}

h1 {
    color: green;
    font-weight: bold;
}

p {
    text-align: center;
    line-height: 1.6;
}

@media (max-width:
600px) {
    .wrapper {
        padding: 15px;
        margin: 15px;
    }
}
```

LANDING PAGE CSS

```
@import
url('https://fonts.googlea
pis.com/css?family=Ral
eway');

*{
    box-sizing: border-
box;
}
.color{
    background: linear-
gradient(to right,
#e1e0ff, #d7fadd);
}

body{
    background: #191919;
    background: green;
    user-select: none;
    letter-spacing: 3px;
    color: #b9b9b9;
    overflow: hidden;
}
p{
    padding-bottom:
15px;
}
p:last-of-type{
    padding-bottom: 0;
}

.wrapper {
    position: absolute;
    left: 20%;
    top: 30px;
    width: 100%;
    height: 100vh;
    display: flex;
    flex-direction:
column;
    align-items: center;
    justify-content:
center;
}

.background-image {
    width: auto;
    height: 700px;
    margin: 25px 10px
10px 10px;
    position: absolute;
    z-index: -1;
    left: 100px;
}

.logo {
    width: 100px;
    position: absolute;
    top: 200px;
```

```
    left: 70%;
    transform:
translateX(-50%);
    z-index: 2;
}

.button{
    cursor: pointer;
    position: relative;
    z-index: 2;
    -webkit-appearance:
none;
    appearance: none;
    background: none;
    outline: none;
    border: none;
    text-transform:
uppercase;
    letter-spacing: 3px;
    border-radius: 2px;
    transition:    all    0.5s
ease;
    font-weight: 600;
    font-size: 14px;
    display: flex;
    align-items: center;
    justify-content:
center;
}

.button-text,  .button-text
p {
    display: inline-block;
    position: relative;
    z-index: 2;
    background:  -webkit-
linear-gradient(to   right,
#FFD700, #4cfa63);
    background:    linear-
gradient(to          right,
#FFD700, #4cfa63);
    -webkit-background-
clip: text;
    background-clip: text;
    color: transparent;
    font-size: 60px;
    text-align: center;
}

.button-text p {
    font-size: 25px;
    display: flex;
    justify-content:
center;
    align-items: center;
    color: white;
}

.button-text h1{
    padding-bottom:
20px;
    text-transform:
uppercase;
    letter-spacing: 3px;
    font-size: 50px;
    color: white;
}

.button:hover     .button-
text p,
.button:hover     .button-
text h1 {
    color: transparent;
}

.button-style2 {
    background-color:
#FFD700;
    color: #ffffff;
    border: none;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    border:    1px    solid
rgba(0,0,0,1);
    border-radius: 5px;
    transition:
background-color 0.3s;
    text-decoration: none;
}

.button-style2:hover {
    background-color:
#ffffff;
    color: #FFD700;
}

.button-style2 p {
    margin: 0;
}

.button-circle{
    position: absolute;
    left: 0;
    top: 0;
    height: 6px;
    width: 6px;
    border-radius: 50%;
    box-shadow: 0 0 10px
#FFD700,   0   0   20px
#FFD700,   0   0   30px
#FFD700;
    transition:    all    0.5s
ease;
}

.button-circle2{
    top: auto;
    bottom: 0;
}

.button-circle3{
    left: auto;
    right: 0;
    box-shadow: 0 0 10px
#4cfa63,    0    0    20px
#4cfa63,    0    0    30px
#4cfa63;
}

.button-circle4{
    left: auto;
    top: auto;
    bottom: 0;
    right: 0;
    box-shadow: 0 0 10px
#4cfa63,    0    0    20px
```

```
#4cfa63, 0 0 30px
#4cfa63;
}

.popup{
   opacity: 0;
   visibility: hidden;
   height: 400px;
   width: 400px;
   flex-shrink: 0;
   border-radius: 3px;
   position: relative;
   z-index: 3;
   display: flex;
   align-items: center;
   justify-content:
center;
   transition: all 0.2s ease
}

.popup-inside{
   position: absolute;
   left: 0;
   top: 0;
   height: 100%;
   width: 100%;
   overflow: hidden;
   border-radius: 50%;
   box-shadow: 0 0 0
black;
   transition:
         box-shadow 0.5s
ease 0.7s,
         border-radius
0.35s ease 0.7s;
}

.backgrounds{
   position: absolute;
   left: 0;
   top: 0;
   height: 100%;
   width: 100%;
   overflow: hidden;
}

.background{
   --offset: 0;
   position: absolute;
   left: var(--offset);
   right: var(--offset);
   bottom: var(--offset);
   top: var(--offset);
   background: linear-
gradient(to right,
#ffea00, #4cfa63);
   transform: scale(0);
   transition: all 0.5s ease
0s;
   border-radius: 50%;
}

.background2{
   --offset: 10%;
   position: absolute;
   left: var(--offset);
   right: var(--offset);
   bottom: var(--offset);
   top: var(--offset);
   background: linear-
gradient(to right,
#ffdd00, #69fa7f);
   transform: scale(0);
   transition: all 0.5s ease
0.1s;
}

.background3{
   --offset: 20%;
   position: absolute;
   left: var(--offset);
   right: var(--offset);
   bottom: var(--offset);
   top: var(--offset);
   background: linear-
gradient(to right,
#ffd000, #85fa99);
   z-index: 2;
   transition: all 0.5s ease
0.2s;
   border-radius: 20px;
}

.background4{
   --offset: 30%;
   position: absolute;
   left: var(--offset);
   right: var(--offset);
   bottom: var(--offset);
   top: var(--offset);
   background: linear-
gradient(to right,
#ffc300, #a7fab7);
   z-index: 3;
   transition: all 0.5s ease
0.3s;
   border-radius: 20px;
}

.background5{
   --offset: 40%;
   position: absolute;
   left: var(--offset);
   right: var(--offset);
   bottom: var(--offset);
   top: var(--offset);
   background: linear-
gradient(to right,
#ffb700, #c3fad1);
   z-index: 4;
   transition: all 0.5s ease
0.4s;
   border-radius: 20px;
}

.background6{
   --offset: 40%;
   position: absolute;
   left: var(--offset);
   right: var(--offset);
   bottom: var(--offset);
   top: var(--offset);
   background: white;
   z-index: 5;
   transition: all 0.8s ease
0.4s;
   border-radius: 20px;
}
```

```
.content{
    --offset: 0;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    display: flex;
    align-items: center;
    justify-content:
center;
    opacity: 0;
    transition: all 0.35s
ease 0.75s;
    z-index: 10;
    border-radius: 20px;
}

body.active .content{
    opacity: 1;
    transform: none;
}

body.active .popup{
    opacity: 1;
    visibility: visible;
    text-align: center;
    width: 700px;
    height: 500px;
}

body.active .popup-
inside{
    border-radius: 30px;
}

body.active
.background{
    transform: scale(1);
}

body.active
.background6{
    transform: scale(8);
}

.button-style {
    background-color:
#FFD700;
    color: #ffffff;
    border: none;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    border: 1px solid
rgba(0,0,0,1);
    border-radius: 5px;
    transition:
background-color 0.3s;
    text-decoration: none;
}

.button-style:hover {
    background-color:
green;
}

.content-wrapper
button.button-style {
    display: block;
    margin: 0 auto;
}

.content-wrapper{
    margin: 10px;
    color: black;
}

@media only screen and
(max-width: 600px) {
    .logo {
        width: 100px;
        position: absolute;
        top: 20%;
        left: 50%;
        transform:
translate(-50%, -50%);
        z-index: -1;
    }
    .background-image {
        width: 250px;
        height: 400px;
        position: absolute;
        top: 70%;
        left: 47%;
        transform:
translate(-50%, -50%);
        z-index: -1;
    }
    .wrapper {
        height: 100vh;
        overflow: hidden;
        display: flex;
        align-items: center;
        justify-content:
center;
        left: 0%;
        top: -50px;
    }
    .button {
        cursor: pointer;
        position: relative;
        width: 80%;
        height: 50px;
        overflow: hidden;
        left: 0;
        z-index: 1;
    }
    .button-text p {
        font-size: 14px;
        font-weight: bold;
    }
    .button-text h1 {
        font-size: 30px;
        padding-bottom:
10px;
    }
    .popup {
        max-height: 100%;
        max-width: 100%;
        bottom: 70px;
        left: 0;
        right: 0;
    }
    .popup-inside {
        height: 130%;
        overflow: hidden;
```

```
        margin: 10px;
        width: 95%;
    }
    .backgrounds {
        height: 100%;
    }
    .content-wrapper {
        margin: 10px;
        text-align: center;
        padding: 5px;
    }
    h2 {
        font-size: 24px;
        margin-top: 140px;
        padding: 5px;
    }
    p {
        font-size: 14px;
        margin-bottom:
0px;
    }
    .second-part-
paragraph {
        text-align: center;
        margin-bottom:
20px;
    }
    .button-style {
        font-size: 16px;
    }
}
```

ADMINISTRATOR AND DEPARTMENT

LOGIN

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
    <meta     charset="utf-
8">
    <meta
name="viewport"
content="width=device-
width, initial-scale=1">
    <title>Login</title>
    <!--                <link
rel="stylesheet"
type="text/css"
href="frontstyle.css">  --
>
    <link
href="https://cdn.jsdeliv
r.net/npm/bootstrap@5.
3.2/dist/css/bootstrap.mi
n.css"    rel="stylesheet"
integrity="sha384-
T3c6CoIi6uLrA9TneNE
oa7RxnatzjcDSCmG1M
XxSR1GAsXEV/Dwwy
kc2MPK8M2HN"
crossorigin="anonymou
s">
    <script
src="https://cdn.jsdelivr.
net/npm/bootstrap@5.3.
2/dist/js/bootstrap.bundl
e.min.js"
integrity="sha384-
C6RzsynM9kWDrMNe
T87bh95OGNyZPhcTN
Xj1NW7RuBCsyN/o0jl
pcV8Qyq46cDfL"
crossorigin="anonymou
s"></script>

    <link  rel="stylesheet"
href="{%            static
'CSS/login.css' %}">
    <!--    favicon    <dev
elgene -->
    <link          rel="icon"
href="{%            static
'elgene.png'          %}"
type="image/x-icon">
    <link      rel="shortcut
icon"  href="{%  static
'favicon.ico'          %}"
type="image/x-icon">
</head>
<body>
    <img  src="{%  static
'layadiwa.png'        %}"
alt="Background Image"
class="background-
image">
    <img  src="{%  static
'elgene.png'          %}"
alt="Logo"
class="logo">
    <div
class="wrapper">
        <span
class="button-text">
            <p>Welcome
to</p>

<h1>VisiThoughts</h1>
        </span>
    </div>
    <div
class="wrapper">
        <div class="button"
id="loginButton"
onclick="document.bod
y.classList.add('active')"
>
            <span
class="button-style2">
                <p>Click   Me
To Login</p>
            </span>
        </div>
    </div>

    <div
class="wrapper">
        <div
class="popup">
            <div
class="popup-inside">
                <div
class="backgrounds">
```

```
                <div
class="background"></d
iv>
                <div
class="background
background2"></div>
                <div
class="background
background3"></div>
                <div
class="background
background4"></div>
                <div
class="background
background5"></div>
                <div
class="background
background6"></div>
            </div>
        </div>
        <div
class="content">
            <form
method="post"
action="{% url 'login'
%}">
                {%
csrf_token %}
                <div>
                    <label
for="username">Userna
me:</label>
                    <input
type="text"
id="username"
name="username"
required>
                </div>
                <div>
                    <label
for="password">Passwo
rd:</label>
                    <input
type="password"
id="password"
name="password"
required>
                </div>
                <button
type="submit">Login</
button>
            </form>
        </div>
    </div>
  </div>
  <script>
    // Get the element
by its ID
    var loginButton =
document.getElementBy
Id('loginButton');

    // Add a click event
listener to the element

loginButton.addEventLi
stener('click', function()
{
        // Hide the button
by setting its display
property to "none"

loginButton.style.displa
y = 'none';
        // Add the 'active'
class to the body
(assuming this is part of
your intended
functionality)

document.body.classList
.add('active');
    });
  </script>
</body>
</html>
```

ADMIN DASH BOARD

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-
8">
  <meta
name="viewport"
content="width=device-
width, initial-
scale=1.0">
  <title>Admin
Dashboard</title>

  <!-- Boothstrap
Online -->
  <link
href="https://cdn.jsdeliv
r.net/npm/bootstrap@5.
3.2/dist/css/bootstrap.mi
n.css" rel="stylesheet"
integrity="sha384-
T3c6CoIi6uLrA9TneNE
oa7RxnatzjcDSCmG1M
XxSR1GAsXEV/Dwwy
kc2MPK8M2HN"
crossorigin="anonymou
s">
  <script
src="https://cdn.jsdelivr.
net/npm/bootstrap@5.3.
2/dist/js/bootstrap.bundl
e.min.js"
integrity="sha384-
C6RzsynM9kWDrMNe
T87bh95OGNyZPhcTN
Xj1NW7RuBCsyN/o0jl
pcV8Qyq46cDfL"
crossorigin="anonymou
s"></script>

  <!-- Link para
lumabas yung pie chart
online -->
  <script
src="https://cdn.jsdelivr.
```

```
net/npm/chart.js"></scri
pt>

    <!-- pang icon ng
account online -->
    <link rel="stylesheet"
href="https://cdnjs.cloud
flare.com/ajax/libs/font-
awesome/6.4.0/css/all.m
in.css" />

    <script
src="https://code.jquery.
com/jquery-
3.6.0.min.js"></script>

    <!-- for history tools
<dev elgene> -->
    <link rel =
"stylesheet" type =
"text/css" href =
"https://cdn.datatables.n
et/1.12.0/css/jquery.data
Tables.min.css"/>
    <meta charset="UTF-
8" name="viewport"
content="width=device-
width, initial-scale=1"/>

    <!-- Add this in the
head section of your
HTML template <dev
elgene> for the
graphofeachdepartmentr
ecordcount-->
    <script
src="https://cdn.jsdelivr.
net/npm/chart.js"></scri
pt>

    <!-- Link of CSS and
JS -->
    <link rel="stylesheet"
href="{% static
'CSS/AdminDashboard.
css' %}">
    <script src="{% static
'JS\AdminDashboard.js'
%}"></script>

    <!-- favicon <dev
elgene -->
    <link rel="icon"
href="{% static
'elgene.png' %}"
type="image/x-icon">
    <link rel="shortcut
icon" href="{% static
'favicon.ico' %}"
type="image/x-icon">
</head>
<body id="data-
container">

<div class="main-
container">


        <div
class="account-icon"
id="account-
dropdown">
            <i class="fas fa-
user"></i>
            {{
request.user.username
}}
            <div
class="dropdown-
content">
                <a href="#"
id="setting"
onclick="openUsername
PasswordModal()">Setti
ng</a>
                <a href="#"
id="exportButton">Exp
ort Data</a>
                <a href="{%
url 'show' %}"
id="manage" >Manage
Staff</a>
                <a href="{%
url 'showdepartment'
%}" id="manage"
>Manage
Department</a>
                <a href="{%
url 'logout' %}"
id="logout">Logout</a
>
            </div>
        </div>
        <!-- <dev elgene> -
->
        <div
id="myModal"
class="modal">
            <div
class="modal-content">
                <span
class="close">&times;<
/span>
                <h2>Change
Username or
Password</h2>
                <p>Select an
option to continue:</p>
                <button
id="changeUsername"
onclick="openChangeU
sernameModal()">Chan
ge Username</button>
                <button
id="changePassword"
onclick="openChangePa
sswordModal()">Chang
e Password</button>
            </div>
        </div>
        <div
id="changeUsernameM
odal" class="modal2">
            <div
class="modal-content">
                <span
class="close">&times;<
/span>
```

```
            <h2>Change
Username</h2>
            <p>Enter your
current username and
password:</p>
            <input
type="text"
id="currentUsername"
placeholder="Current
Username" required>
            <input
type="password"
id="currentPassword"
placeholder="Current
Password" required>
            <p>Enter your
new username:</p>
            <input
type="text"
id="newUsername"
placeholder="New
Username" required>
            <button
id="saveUsername"
onclick="saveUsername
()">Save</button>
          </div>
        </div>
        <div
id="changePasswordMo
dal" class="modal2">
          <div
class="modal-content">
            <span
class="close">&times;<
/span>
            <h2>Change
Password</h2>
            <p>Enter your
current username and
password:</p>
            <input
type="text"
id="currentUsername1"
placeholder="Current
Username" required>
            <input
type="password"
id="currentPassword1"
placeholder="Current
Password" required>
            <p>Enter your
new password:</p>
            <input
type="password"
id="newPassword"
placeholder="New
Password" required>
            <button
id="savePassword"
onclick="savePassword(
)">Save</button>
          </div>
        </div>

        <!-- JavaScript code
<dev elgene> -->
        <script>
          function
openUsernamePassword
Modal() {
            var modal =
document.getElementBy
Id("myModal");
            var close =
document.getElementsB
yClassName("close")[0]
;


modal.style.display =
"block";

            close.onclick =
function() {

modal.style.display =
"none";
            }
          }

          function
openChangeUsernameM
odal() {
            var modal1 =
document.getElementBy
Id("changeUsernameMo
dal");
            var close1 =
document.getElementsB
yClassName("close")[1]
;
            var modal =
document.getElementBy
Id("myModal");


modal.style.display =
"none";

modal1.style.display =
"block";

            close1.onclick
= function() {

modal1.style.display =
"none";

modal.style.display =
"block";
            }
          }

          function
openChangePasswordM
odal() {
            var modal2 =
document.getElementBy
Id("changePasswordMo
dal");
            var close2 =
document.getElementsB
yClassName("close")[2]
;
```

```
var modal = document.getElementById("myModal");

modal.style.display = "none";
modal2.style.display = "block";

close2.onclick = function() {
modal2.style.display = "none";
modal.style.display = "block";
}
}
function saveUsername() {
var currentUsername = $("#currentUsername").val();
var newUsername = $("#newUsername").val();
var currentPassword = $("#currentPassword").val();
$.ajax({
type: 'POST',
url: "{% url 'change_username' %}",
data: {
'current_username': currentUsername,
'new_username': newUsername,
'current_password': currentPassword,
'csrfmiddlewaretoken': $('input[name=csrfmiddlewaretoken]').val()
},
success: function (response) {
if (response.success) {
alert('Username successfully changed');
} else {
alert('Error: ' + response.error);
}
},
error: function (xhr, status, error) {
alert('Error: ' + error);
}
});
}
// <dev elgene>
function savePassword() {
var currentUsername = $("#currentUsername1").val();
var currentPassword = $("#currentPassword1").val(); // Updated the ID
var newPassword = $("#newPassword").val();

$.ajax({
type: 'POST',
url: "{% url 'change_password' %}",
data: {
'current_username': currentUsername,
'current_password': currentPassword,
'new_password': newPassword,
'csrfmiddlewaretoken': $('input[name=csrfmiddlewaretoken]').val()
},
success: function (response) {
if (response.success) {
alert('Password successfully changed');
} else {
// Display the error message received from the server
alert('Error: ' + JSON.stringify(response.error));
}
},
error: function (xhr, status, error) {
```

```
alert('Error: ' + error);
                }
            });
        }
    </script>

  <div class="center-text">
    <h1>Admin Dashboard</h1>
  </div>


  <!--filtering tools <dev elgene>-->
  <form class="tools">
    {% csrf_token %}
    <div class="select-staff-container" id="AdminDashboard">
        <!-- Staff filter -->
        <select class="each-DEpartment" id="eachDepartment" name="eachDepartment">
            <option value="">Over All Evaluation</option>
            {% for department in department_names %}
                <option value="{{ department.id }}" {% if department.id|stringformat:"s" == selected_department_id %}selected{% endif %}>{{ department.name }}</option>
            {% endfor %}
        </select>
        <select class="select-staff" id="attendingStaff" name="attendingStaff">
            <option value="">{{ selected_staff_name }}</option>
            {% for staff in staffs %}
                <option value="{{ staff.pk }}">{{ staff.name }}</option>
            {% endfor %}
        </select>
        <script>

$("#eachDepartment").change(function () {
            const url = "{% url 'ajax_load_staffs' %}";
            const departmentId = $(this).val();

            if (departmentId === '') {
                // Clear the content of the "attendingStaff" dropdown when "CCAT Over All" is selected

$("#attendingStaff").html('<option value="">None</option>');
            } else {
                $.ajax({
                    url: url,
                    data: {
                        'department_id': departmentId
                    },

                    success: function (data) {

$("#attendingStaff").html(data);
                    }
                });
            }
        });
        </script>
    </div>
    <!-- Date filter -->
    <div class="date-filter">
        <label for="start-date">From:</label>
        <input type="date" id="start-date" name="start-date" value="{{ start_date|default:'' }}">
        <label for="end-date">To:</label>
        <input type="date" id="end-date" name="end-date" value="{{ end_date|default:'' }}">
    </div>
    <button type="button" id="fetchDataButton" onclick="handleButtonClick()">Over All Analysis</button>
  </form>
  <!--filtering tools <dev elgene>-->

  <!-- Areas of Concern <dev elgene>-->
```

```
<div class="left-container">
    <h1>Areas of Concern</h1>
    <h2>
        {% if selected_staff_name != "None" %}
            {{ selected_staff_name }}
        {% else %}
            {{ selected_department_name }}
        {% endif %}
    </h2>

    <div class="receipt-details">
        <div class="item">
            A. Courtesy
            <span>{{ average_courtesy|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            B. Service
        </div>
        <div class="item">
            <span>1. Quality</span>
            <span>{{ average_quality|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            <span>2. Timeliness</span>
            <span>{{ average_timeless|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            <span>3. Efficiency</span>
            <span>{{ average_efficiency|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            C. Physical Condition of Office / Work Space
        </div>
        <div class="item">
            <span>1. Cleanliness</span>
            <span>{{ average_cleanliness|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            <span>2. Comfort</span>
            <span>{{ average_comfort|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
    </div>

    <div class="total">
        <span>Total Average:</span>
        <span>{{ total_average|default_if_none:"N/A"|floatformat:2 }}%</span>
    </div>
</div>
<!--end of areas of concern <dev elgene>-->


<!--pie chart for sentiment <dev elgene>-->
<div class="pie-chart-container">
    <div class="chart-title">
        Sentiment Results <br>
        Total Sentiments: {{ total_sentiments }}
    </div>
    <canvas id="sentimentPieChart"></canvas>
</div>
<!--pie chart for sentiment <dev elgene>-->

<!-- Add this where you want the chart to appear in your HTML template -->
<div class="departmentcount-container">
    <h2>Department Records</h2>
    <p>Feedback Records of Each Department</p>
    <canvas id="departmentChart">
</canvas>
</div>

<div class="table-container">
```

```
<div class="table-wrapper">
<table class="fl-table" id="bootstrapdatatable">
<thead>
<tr>
<th>ID</th>
<th>Client Name</th>
<th>Client Agency</th>
<th>Email</th>
<th>Purpose of Visit</th>
<th>Date of Visit</th>
<th>Concerned Department</th>
<th>Attending Staff</th>
<th>Courtesy</th>
<th>Quality</th>
<th>Timeliness</th>
<th>Efficiency</th>
<th>Cleanliness</th>
<th>Comfort</th>
<th>Comment</th>
<th>Sentiment</th>
</tr>
</thead>
<tbody>
{% for feedback in feedback_records %}
<tr>
<td>{{ feedback.id }}</td>
<td>{{ feedback.name }}</td>
<td>{{ feedback.agency }}</td>
<td>{{ feedback.email }}</td>
<td>{{ feedback.purposeOfVisit }}</td>
<td>{{ feedback.dateOfVisit }}</td>
<td>{{ feedback.department }}</td>
<td>{{ feedback.staff }}</td>
<td>{{ feedback.courtesy}}</td>
<td>{{ feedback.quality}}</td>
<td>{{ feedback.timeless}}</td>
<td>{{ feedback.efficiency}}</td>
<td>{{ feedback.cleanliness}}</td>
<td>{{ feedback.comfort}}</td>
<td>{{ feedback.comment }}</td>
<td>{{ feedback.sentiment }}</td>
<!-- Add other fields you want to display <develgene> -->
</tr>
{% empty %}
<tr>
<td colspan="16">No records found.</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
<!-- Add this script at the end of your HTML file or in a <script> tag -->
<script>
// Get the data from the Django context
var departmentNames = {{ department_names_for_plot|safe }};
var positiveSentiments = {{ positive_sentiments|safe }};
var neutralSentiments = {{ neutral_sentiments|safe }};
var negativeSentiments = {{ negative_sentiments|safe }};
var totalRecords = {{
```

```
record_counts_for_plot|safe }};

    // Prepare the data for the chart
    var data = {
        labels: departmentNames,
        datasets: [{
            label: 'Positive Sentiments',
            data: positiveSentiments,
            backgroundColor: 'rgba(75, 192, 192, 1)',
            borderColor: 'rgba(75, 192, 192, 1)',
            borderWidth: 1,
            stack: 'stack1' // Specify stack name for stacking
        }, {
            label: 'Neutral Sentiments',
            data: neutralSentiments,
            backgroundColor: 'rgba(255, 206, 86, 1)',
            borderColor: 'rgba(255, 206, 86, 1)',
            borderWidth: 1,
            stack: 'stack1'
        }, {
            label: 'Negative Sentiments',
            data: negativeSentiments,
            backgroundColor: 'rgba(255, 99, 132, 1)',
            borderColor: 'rgba(255, 99, 132, 1)',
            borderWidth: 1,
            stack: 'stack1'
        }]
    };

    // Get the canvas element
    var ctx = document.getElementById('departmentChart').getContext('2d');

    // Create the chart
    var myBarChart = new Chart(ctx, {
        type: 'bar',
        data: data,
        options: {
            scales: {
                y: {
                    beginAtZero: true,
                    stacked: true // Enable stacking for the y-axis
                }
            },
            responsive: true,
            maintainAspectRatio: false,
            plugins: {
                legend: {
                    display: true
                },
                tooltip: {
                    callbacks: {
                        label: function(context) {
                            var departmentIndex = context.dataIndex;
                            var total = totalRecords[departmentIndex];
                            var positive = positiveSentiments[departmentIndex];
                            var neutral = neutralSentiments[departmentIndex];
                            var negative = negativeSentiments[departmentIndex];

                            return 'Positive: ' + positive + ' | Neutral: ' + neutral + ' | Negative: ' + negative + ' | Total: ' + total;
                        }
                    }
                }
            }
        }
    });
</script>

<!-- for each department record <dev elgene> -->
<script src = "https://cdn.datatables.net/1.12.0/js/jquery.dataTables.min.js"></script>
<script>
$(document).ready(function() {

$('#bootstrapdatatable').DataTable({
    "aLengthMenu": [[5, 10, 20, 50, -1], [5, 10, 20, 50, "All"]],
```

```
        "iDisplayLength": 5,
        "order": [[0, 'desc']] // Assuming the first column is the ID column
      }
    );
} );
</script>
<script>
    function fetchData() {
        var selectedDepartment = $('#eachDepartment').val();
        var selectedStaff = $("#attendingStaff").val();
        var startDate = $("#start-date").val(); // Get the selected start date
        var endDate = $("#end-date").val(); // Get the selected end date

        $.ajax({
            type: "POST",
            url: "{% url 'AdminDashboard' %}", // Replace 'your_view_name' with the actual URL pattern name for your view
            data: {
                'department':selectedDepartment,
                'attendingStaff': selectedStaff,
                'start-date': startDate, // Pass the selected start date
                'end-date': endDate, // Pass the selected end date
                csrfmiddlewaretoken: "{{ csrf_token }}"
            },
            success: function(data) {
                $("#data-container").html(data);
            }
        });
    }

    // handle the button text details
    function updateButtonText() {
        var selectedStaff = $("#attendingStaff").val();
        var startDate = $("#start-date").val();
        var endDate = $("#end-date").val();
        var selectedDepartment = $('#eachDepartment').val();
        var buttonText = "Over All Analysis";

        if (selectedStaff || selectedDepartment || (startDate && endDate)) {
            buttonText = "Custom Analysis";
        }

        $("#fetchDataButton").text(buttonText);
    }

    // Handle change event of the staff
    $("#attendingStaff").on("change", function () {
        updateButtonText();
    });
    // Handle change event of the department
    $("#eachDepartment").on("change", function () {
        updateButtonText();
    });

    // Handle change event of the date inputs
    $("#start-date, #end-date").on("change", function () {
    if ($("#start-date").val() && $("#end-date").val()) {
        updateButtonText();
    }
});

    function resetFilters() {
        // Check if the attending staff dropdown has a value selected
        if ($("#eachDepartment").val() !== "") {
            $("#eachDepartment").val(""); // Reset the dropdown to "Over All"
        }
```

```
        // Check if the start date and end date inputs are not empty
        if ($("#start-date").val() !== "" || $("#end-date").val() !== "") {
            $("#start-date").val(""); // Clear the start date
            $("#end-date").val(""); // Clear the end date
        }
    }

    function handleButtonClick() {
        var buttonText = $("#fetchDataButton").text();

        if (buttonText === "Over All Analysis") {
            resetFilters(); // Call resetFilters only when the button text is "Over All Analysis"
        }

        fetchData(); // Always call fetchData
    }

    var sentimentCounts = {{ sentiment_counts|safe }};
    var sentimentLabels = Object.keys(sentimentCounts);
    var sentimentData = Object.values(sentimentCounts);
    var totalSentiments = sentimentData.reduce((a, b) => a + b, 0);
    // Calculate percentages
    var sentimentPercentages = sentimentData.map(count => ((count / totalSentiments) * 100).toFixed(2));
    // Create labels with percentages
    var labelsWithPercentages = sentimentLabels.map((label, index) => `${label}: ${sentimentData[index]} (${sentimentPercentages[index]}%)`);
    var sentimentPieChartCanvas = document.getElementById('sentimentPieChart').getContext('2d');
    var sentimentPieChart = new Chart(sentimentPieChartCanvas, {
        type: 'pie',
        data: {
            labels: labelsWithPercentages, // Use the labels with percentages
            datasets: [{
                data: sentimentData,
                backgroundColor: ['rgba(75, 192, 192, 1)', 'rgba(255, 206, 86, 1)', 'rgba(255, 99, 132, 1)'],
                borderColor: ['rgba(75, 192, 192, 1)', 'rgba(255, 206, 86, 1)', 'rgba(255, 99, 132, 1)'],
                borderWidth: 1
            }]
        },
        options: {
            responsive: true,
            plugins: {
                legend: {
                    display: true,
                    position: 'bottom',
                    aligned: 'start',
                },
                tooltips: {
                    callbacks: {
                        label: function (tooltipItem, data) {
                            return data.labels[tooltipItem.index]; // Show labels with percentages in tooltips
                        }
                    }
                }
            }
        }
    });

    function decodeHtmlEntities(text) {
        var txt = document.createElement("textarea");
        txt.innerHTML = text;
        return txt.value;
    }

    // <dev elgene>

document.getElementBy
```

```
Id("exportButton").addEventListener("click", function() {
    // Create a JSON object with your data
    var data = `DEPARTMENT NAME: ${"{{selected_department_name}}"}\n` +
    `Staff ID: ${"{{selected_staff_id}}"}\n` +
    `Staff Name: ${"{{selected_staff_name}}"}\n` +
    `Start Date: ${"{{start_date}}"}\n` +
    `End Date: ${"{{end_date}}"}\n\n` +
    `Average Courtesy: ${parseFloat("{{average_courtesy}}").toFixed(2)}\n` +
    `Average Quality: ${parseFloat("{{average_quality}}").toFixed(2)}\n` +
    `Average Timeliness: ${parseFloat("{{average_timeless}}").toFixed(2)}\n` +
    `Average Efficiency: ${parseFloat("{{average_efficiency}}").toFixed(2)}\n` +
    `Average Cleanliness: ${parseFloat("{{average_cleanliness}}").toFixed(2)}\n` +
    `Average Comfort: ${parseFloat("{{average_comfort}}").toFixed(2)}\n\n` +
    `Total Average: ${parseFloat("{{total_average}}").toFixed(2)}\n` +
    `Total Records: ${"{{total_records}}"}\n` +
    `Positive Sentiment: ${sentimentCounts['Positive Sentiment']}\n` +
    `Neutral Sentiment: ${sentimentCounts['Neutral Sentiment']}\n` +
    `Negative Sentiment: ${sentimentCounts['Negative Sentiment']}\n\n`;

    var departmentNames = {{ department_names_for_plot|safe }};
    var recordCounts = {{ record_counts_for_plot|safe }};
    var positiveSentiments = {{ positive_sentiments|safe }};
    var negativeSentiments = {{ negative_sentiments|safe }};
    var neutralSentiments = {{ neutral_sentiments|safe }};

console.log("Department Names:", departmentNames);
console.log("Record Counts:", recordCounts);

console.log("Positive Sentiments:", positiveSentiments);

console.log("Negative Sentiments:", negativeSentiments);

console.log("Neutral Sentiments:", neutralSentiments);

    for (var i = 0; i < departmentNames.length; i++) {
        var departmentName = departmentNames[i];
        var recordCount = recordCounts[i];
        var positiveSentiment = positiveSentiments[i];
        var negativeSentiment = negativeSentiments[i];
        var neutralSentiment = neutralSentiments[i];

        data += `${departmentName}: Total Records: ${recordCount || 0}, Positive: ${positiveSentiment || 0}, Negative: ${negativeSentiment || 0}, Neutral: ${neutralSentiment || 0}\n`;
    }
```

```
        data = decodeHtmlEntities(data);

        // Check and set values to "None" if they are empty strings
        for (var key in data) {
            if (data[key] == "") {
                data[key] = "None";
            }
        }

        var blob = new Blob([data], { type: "text/plain" });

        // Create a link element to trigger the download
        var a = document.createElement("a");
        a.href = URL.createObjectURL(blob);
        a.download = "Evaluation_Report.txt";

        // Simulate a click on the link to initiate the download
        a.click();
    });
</script>
</body>
</html>
```

STAFF DROPDOWN

```
{% for staff in staffs %}
<option value="{{ staff.pk }}">{{ staff.name }}</option>
{% endfor %}
```

BASE HTML

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">
<title>{% block title %}{% endblock %}</title>
<!-- <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"/> -->
<link rel = "stylesheet" type = "text/css" href = "https://cdn.datatables.net/1.12.0/css/jquery.dataTables.min.css"/>
<link rel="stylesheet" type="text/css" href="{% static 'CSS/crud.css' %}">
<meta charset="UTF-8" name="viewport" content="width=device-width, initial-scale=1"/>
<!-- favicon <dev elgene -->
<link rel="icon" href="{% static 'elgene.png' %}" type="image/x-icon">
<link rel="shortcut icon" href="{% static 'favicon.ico' %}" type="image/x-icon">
<!-- Add these lines to your head section in base.html or the template <dev elgene> -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.3/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>
<body>
<div class="main-container">
 <div class="row">
  <div class="center-text">
    <h1>Staff Management</h1>
  </div>
  {% block content %}{% endblock %}
 </div>
</div>

<script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.
```

```
2/dist/js/bootstrap.bundle.min.js"></script>
<script src = "https://cdn.datatables.net/1.12.0/js/jquery.dataTables.min.js"></script>
<script>
$(document).ready(function() {

$('#bootstrapdatatable').DataTable({
    "aLengthMenu": [[10, 20, 30, 40, 50, -1], [10, 20, 30, 40, 50, "All"]],
     "iDisplayLength": 10
    }
  );
} );
</script>
</body>
</html>
```

EDIT HTML

```
{% extends "base.html" %}

{% block title %}Edit Staff{% endblock title %}

{% block content %}
<div class="col-md-12-edit">
  <form method="POST" class="post-form" action="{% url 'update' employee.id %}">
      {% csrf_token %}
  <div class="container">
      <div class="col-12">
          <h1>Update Details</h1>
      </div>
      <div class="mb-3">

<label>Name:</label>
          <input type="text" class="form-control" name="name" id="id_name" required maxlength="100" value="{{ employee.name }}" />
      </div>
      <div class="mb-3">
          <label for="id_department">Department:</label>
          <select class="form-select" name="department" id="id_department" required>
              <option value="" {% if not staff_department %}selected{% endif %}>Choose a department</option>
              {% for department in department_names %}
                  <option value="{{ department.id }}" {% if staff_department and staff_department.id == department.id %}selected{% endif %}>
                      {{ department.name }}
                  </option>
              {% endfor %}
          </select>
      </div>
      <div class="mb-3 button-container">
          <span class="btn-add-new-2">
              <a href="{% url 'show' %}" style="font-weight: normal; font-family:'Times New Roman', Times, serif; font-size: 15px;">Back</a>
          </span>
          <button type="submit" class="btn btn-success btn-lg" style="font-weight: normal; font-family:'Times New Roman', Times, serif; font-size: 15px;">Update</button>
      </div>
  </div>
  </form>
{% endblock content %}
```

INDEX HTML

```
{% extends "base.html" %}

{% block title %}Add New Staff{% endblock title %}

{% block content %}
<div class="col-md-12-add">
  <form method="POST"
```

```
class="post-form" action="/departmentsdashboard/addnew">
  {% csrf_token %}
  <div class="container">
    <div class="col-12">
    <h3>Enter Staff Details</h3>
    </div>

    <div class="mb-3">
      <label for="fullName" class="form-label">Enter Full Name:</label>
      {{ form.name }}
    </div>
    <div class="mb-3">
      <label for="concernedDepartment" class="form-label">Select Department:</label>
      {{ form.department }}
    </div>

    <div class="mb-3 button-container">
      <span class="btn-add-new-2">
        <a href="{% url 'show' %}" style="font-weight: normal; font-family:'Times New Roman', Times, serif; font-size: 15px;">Back</a>
      </span>
      <button type="submit" class="btn btn-success btn-lg" style="font-weight: normal; font-family:'Times New Roman', Times, serif; font-size: 15px;">Submit</button>
    </div>
  </div>
  </form>
</div>
{% endblock content %}
```

SHOW HTML

```
{% extends "base.html" %}

{% block title %}Manage Staffs{% endblock title %}

{% block content %}
  <div class="col-md-12">
    <div class="button-container">
      <span class="btn-add-new">
        <a href="{% url 'AdminDashboard' %}">HOME</a>
        <a href="{% url 'addnew' %}">Add New Staff</a>
      </span>
    </div>
    <div class="table-responsive">
      <table id="bootstrapdatatable" class="table table-striped table-bordered" width="100%">
        <thead>
        <th>ID</th>

<th>Name</th>

<th>Department</th>

<th>Action</th>
        </thead>
        <tbody>
        {% for staff in staffs %}
          <tr>
            <td>{{ staff.id }}</td>
            <td>{{ staff.name }}</td>
            <td>{{ staff.department_name }}</td>
            <td>
              <div style="display: flex; flex-flow: row; align-items: center; justify-content: center; flex-wrap: wrap; width: 100%;">
                <a href="{% url 'edit' staff.id %}" class="btn btn-primary">Edit</a>
                <div style="margin: 0 2px 0 2px;"> </div>

<button class="btn btn-danger" data-toggle="modal" data-target="#deleteModal{{ staff.id }}">Delete</button>
              </div>
            </td>
```

```
                </tr>
                <!-- Modal -->
                <div class="modal fade" id="deleteModal{{ staff.id }}" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
                    <div class="modal-dialog" role="document">
                        <div class="modal-content">
                            <div class="modal-header">

<h5 class="modal-title" id="deleteModalLabel">Confirm Delete</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>
                            <div class="modal-body">

Are you sure you want to delete {{ staff.name }}?

</div>
                            <div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>

<a href="{% url 'delete' staff.id %}" class="btn btn-danger">Delete</a>

</div>
                        </div>
                    </div>
                </div>
                <!-- End Modal -->
            {% endfor %}
            </tbody>
        </table>
    </div>
  </div>
{% endblock content %}
```

DEPARTMENT DASHBOARD

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Department Dashboard</title>

  <!-- Boothstrap Online -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL" crossorigin="anonymous"></script>

  <!-- Link para lumabas yung pie chart online -->
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

  <!-- pang icon ng account online -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" />

  <!-- for history tools <dev elgene> -->
  <link rel = "stylesheet" type = "text/css" href = "https://cdn.datatables.net/1.12.0/css/jquery.dataTables.min.css"/>
  <script src="https://code.jquery.
```

```
com/jquery-3.6.0.min.js"></script>

    <!-- Link of CSS and JS -->
    <link rel="stylesheet" href="{% static 'CSS/dashboardstyle.css' %}">
    <script src="{% static 'JS\dashboardscript.js' %}"></script>

    <!-- favicon <dev elgene -->
    <link rel="icon" href="{% static 'elgene.png' %}" type="image/x-icon">
    <link rel="shortcut icon" href="{% static 'favicon.ico' %}" type="image/x-icon">
</head>
<body id="data-container">

    <div class="main-container">

        <div class="account-icon" id="account-dropdown">
            <i class="fas fa-user"></i>
            {{ request.user.username }}
            <div class="dropdown-content">
                <a href="#" id="setting" onclick="openUsernamePasswordModal()">Setting</a>
                <a href="{% url 'logout' %}" id="logout">Logout</a>
            </div>
        </div>
        <!-- <dev elgene> -->
        <div id="myModal" class="modal">
            <div class="modal-content">
                <span class="close">&times;</span>
                <h2>Change Username or Password</h2>
                <p>Select an option to continue:</p>
                <button id="changeUsername" onclick="openChangeUsernameModal()">Change Username</button>
                <button id="changePassword" onclick="openChangePasswordModal()">Change Password</button>
            </div>
        </div>
        <div id="changeUsernameModal" class="modal2">
            <div class="modal-content">
                <span class="close">&times;</span>
                <h2>Change Username</h2>
                <p>Enter your current username and password:</p>
                <input type="text" id="currentUsername" placeholder="Current Username" required>
                <input type="password" id="currentPassword" placeholder="Current Password" required>
                <p>Enter your new username:</p>
                <input type="text" id="newUsername" placeholder="New Username" required>
                <button id="saveUsername" onclick="saveUsername()">Save</button>
            </div>
        </div>
        <div id="changePasswordModal" class="modal2">
            <div class="modal-content">
                <span class="close">&times;</span>
                <h2>Change Password</h2>
                <p>Enter your current username and password:</p>
                <input type="text" id="currentUsername1" placeholder="Current Username" required>
                <input type="password"
```

```
id="currentPassword1"
placeholder="Current
Password" required>
            <p>Enter your
new password:</p>
            <input
type="password"
id="newPassword"
placeholder="New
Password" required>
            <button
id="savePassword"
onclick="savePassword(
)">Save</button>
        </div>
    </div>

    <!-- JavaScript code
<dev elgene> -->
    <script>
        function
openUsernamePassword
Modal() {
            var modal =
document.getElementBy
Id("myModal");
            var close =
document.getElementsB
yClassName("close")[0]
;


modal.style.display =
"block";

            close.onclick =
function() {

modal.style.display =
"none";
            }
        }

        function
openChangeUsernameM
odal() {
            var modal1 =
document.getElementBy
Id("changeUsernameMo
dal");
            var close1 =
document.getElementsB
yClassName("close")[1]
;
            var modal =
document.getElementBy
Id("myModal");


modal.style.display =
"none";

modal1.style.display =
"block";

            close1.onclick
= function() {

modal1.style.display =
"none";

modal.style.display =
"block";
            }
        }

        function
openChangePasswordM
odal() {
            var modal2 =
document.getElementBy
Id("changePasswordMo
dal");
            var close2 =
document.getElementsB
yClassName("close")[2]
;
            var modal =
document.getElementBy
Id("myModal");


modal.style.display =
"none";

modal2.style.display =
"block";

            close2.onclick
= function() {

modal2.style.display =
"none";

modal.style.display =
"block";
            }
        }
        function
saveUsername() {
            var
currentUsername =
$("#currentUsername").
val();
            var
newUsername =
$("#newUsername").val
();
            var
currentPassword =
$("#currentPassword").v
al();
            $.ajax({
                type:
'POST',
                url: "{% url
'change_username' %}",
                data: {

'current_username':
currentUsername,

'new_username':
newUsername,

'current_password':
currentPassword,
```

```
'csrfmiddlewaretoken': $('input[name=csrfmiddlewaretoken]').val()
                },
                success: function (response) {
                    if (response.success) {

alert('Username successfully changed');
                    } else {

alert('Error: ' + response.error);
                    }
                },
                error: function (xhr, status, error) {

alert('Error: ' + error);
                }
            });
        }
        // <dev elgene>
        function savePassword() {
            var currentUsername = $("#currentUsername1").val();
            var currentPassword = $("#currentPassword1").val(); // Updated the ID
            var newPassword = $("#newPassword").val();



            $.ajax({
                type: 'POST',
                url: "{% url 'change_password' %}",
                data: {

'current_username': currentUsername,

'current_password': currentPassword,

'new_password': newPassword,

'csrfmiddlewaretoken': $('input[name=csrfmiddlewaretoken]').val()
                },
                success: function (response) {
                    if (response.success) {

alert('Password successfully changed');
                    } else {
                        // Display the error message received from the server

alert('Error: ' + JSON.stringify(response.error));
                    }
                },
                error: function (xhr, status, error) {

alert('Error: ' + error);
                }
            });
        }
    </script>
```

```
    <div class="center-text">

<h1>{{department.name}}</h1>

<h2>Dashboard</h2>
    </div>


    <!--filtering tools <dev elgene>-->
    <form class="tools">
        {% csrf_token %}
        <div class="select-staff-container">
            <!-- Staff filter -->
            <select class="select-staff" id="attendingStaff" name="attendingStaff">
                <option value="">Over All Evaluation</option>
                {% for staff in staff_members %}
                    <option value="{{ staff.id }}" {% if staff.id|stringformat:"s" == selected_staff_id %}selected{% endif %}>{{ staff.name }}</option>
                {% endfor %}
            </select>

        </div>
        <!-- Date filter -->
        <div class="date-filter">
            <label for="start-date">From:</label>
            <input type="date" id="start-
```

```
date" name="start-date" value="{{ start_date|default:" }}">
        <label for="end-date">To:</label>
        <input type="date" id="end-date" name="end-date" value="{{ end_date|default:" }}">
    </div>
    <button type="button" id="fetchDataButton" onclick="handleButtonClick()">Over All Analysis</button>
  </form>
  <!--filtering tools <dev elgene>-->

  <!-- Areas of Concern <dev elgene>-->
  <div class="left-container">
    <h1>Areas of Concern</h1>

<h2>{{selected_staff_name}}</h2>
    <div class="receipt-details">
        <div class="item">
            A. Courtesy
            <span>{{ average_courtesy|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            B. Service
        </div>
        <div class="item">
            <span>1. Quality</span>
            <span>{{ average_quality|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            <span>2. Timeliness</span>
            <span>{{ average_timeless|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            <span>3. Efficiency</span>
            <span>{{ average_efficiency|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            C. Physical Condition of Office / Work Space
        </div>
        <div class="item">
            <span>1. Cleanliness</span>
            <span>{{ average_cleanliness|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
        <div class="item">
            <span>2. Comfort</span>
            <span>{{ average_comfort|default_if_none:"N/A"|floatformat:2 }}%</span>
        </div>
    </div>

    <div class="total">
        <span>Total Average:</span>
        <span>{{ total_average|default_if_none:"N/A"|floatformat:2 }}%</span>
    </div>
  </div>
  <!--end of areas of concern <dev elgene>-->


  <!--pie chart for sentiment <dev elgene>-->
  <div class="pie-chart-container">
    <div class="chart-title">
        Sentiment Results <br>
        Total Sentiments: {{ total_sentiments }}
    </div>
    <canvas id="sentimentPieChart"></canvas>
  </div>
  <!--pie chart for sentiment <dev elgene>-->


  <div class="table-container">
    <div class="table-wrapper">
        <table class="fl-table"
```

```
id="bootstrapdatatable"
>
            <thead>
              <tr>
<th>ID</th>
<th>Client Name</th>
<th>Client Agency</th>
<th>Email</th>
<th>Purpose of Visit</th>
                <th>Date of Visit</th>
<th>Concerned Department</th>
<th>Attending Staff</th>
<th>Courtesy</th>
<th>Quality</th>
<th>Timeliness</th>
<th>Efficiency</th>
<th>Cleanliness</th>
<th>Comfort</th>
<th>Comment</th>
<th>Sentiment</th>
              </tr>
            </thead>
            <tbody>
              {% for feedback in feedback_records %}
                <tr>
                <td>{{ feedback.id }}</td>
                <td>{{ feedback.name }}</td>
                <td>{{ feedback.agency }}</td>
                <td>{{ feedback.email }}</td>
                <td>{{ feedback.purposeOfVisit }}</td>
                <td>{{ feedback.dateOfVisit }}</td>
                <td>{{ feedback.department }}</td>
                <td>{{ feedback.staff }}</td>
                <td>{{ feedback.courtesy}}</td>
                <td>{{ feedback.quality}}</td>
                <td>{{ feedback.timeless}}</td>
                <td>{{ feedback.efficiency}}</td>
                <td>{{ feedback.cleanliness}}</td>
                <td>{{ feedback.comfort}}</td>
                <td>{{ feedback.comment }}</td>
                <td>{{ feedback.sentiment }}</td>
                <!-- Add other fields you want to display <develgene> -->
                </tr>
              {% empty %}
                <tr>
                  <td colspan="16">No records found.</td>
                </tr>
              {% endfor %}
            </tbody>
          </table>
        </div>
    </div>
</div>
<script src = "https://cdn.datatables.net/1.12.0/js/jquery.dataTables.min.js"></script>
<script>
$(document).ready(function() {

$('#bootstrapdatatable').DataTable({
    "aLengthMenu": [[5, 10, 20, 50, -1], [5, 10, 20, 50, "All"]],
    "iDisplayLength": 5,
    "order": [[0, 'desc']] // Assuming the first column is the ID column
    }
  );
} );
</script>
<script>
  function fetchData() {
    var selectedStaff = $("#attendingStaff").val();
    var startDate = $("#start-date").val(); //
```

```
Get the selected start date
        var endDate = $("#end-date").val(); // Get the selected end date

        $.ajax({
            type: "POST",
            url: "{% url 'DepartmentDashboard' %}",  // Replace 'your_view_name' with the actual URL pattern name for your view
            data: {
                'attendingStaff': selectedStaff,
                'start-date': startDate,  // Pass the selected start date
                'end-date': endDate,  // Pass the selected end date
                csrfmiddlewaretoken: "{{ csrf_token }}"
            },
            success: function(data) {
                $("#data-container").html(data);
            }
        });
    }

    // handle the button text details
    function updateButtonText() {
        var selectedStaff = $("#attendingStaff").val();
        var startDate = $("#start-date").val();
        var endDate = $("#end-date").val();
        var buttonText = "Over All Analysis";

        if ((selectedStaff && !startDate) || (!selectedStaff && startDate) || (selectedStaff && startDate)) {
            buttonText = "Custom Analysis";
        }

        $("#fetchDataButton").text(buttonText);
    }

    // Handle change event of the dropdown
    $("#attendingStaff").on("change", function () {
        updateButtonText();
    });
    // Handle change event of the date inputs
    $("#start-date, #end-date").on("change", function () {
        updateButtonText();
    });

    function resetFilters() {
        // Check if the attending staff dropdown has a value selected
        if ($("#attendingStaff").val() !== "") {
            $("#attendingStaff").val(""); // Reset the dropdown to "Over All"
        }

        // Check if the start date and end date inputs are not empty
        if ($("#start-date").val() !== "" || $("#end-date").val() !== "") {
            $("#start-date").val(""); // Clear the start date
            $("#end-date").val(""); // Clear the end date
        }
    }

    function handleButtonClick() {
        var buttonText = $("#fetchDataButton").text();

        if (buttonText === "Over All Analysis") {
            resetFilters(); // Call resetFilters only when the button text is "Over All Analysis"
        }

        fetchData(); // Always call fetchData
    }

    var sentimentCounts = {{ sentiment_counts|safe }};
    var sentimentLabels = Object.keys(sentimentCounts);
```

```
var sentimentData = Object.values(sentimentCounts);
var totalSentiments = sentimentData.reduce((a, b) => a + b, 0);

// Calculate percentages
var sentimentPercentages = sentimentData.map(count => ((count / totalSentiments) * 100).toFixed(2));

// Create labels with percentages
var labelsWithPercentages = sentimentLabels.map((label, index) => `${label}: ${sentimentData[index]} (${sentimentPercentages[index]}%)`);

var sentimentPieChartCanvas = document.getElementById('sentimentPieChart').getContext('2d');
var sentimentPieChart = new Chart(sentimentPieChartCanvas, {
    type: 'pie',
    data: {
        labels: labelsWithPercentages, // Use the labels with percentages
        datasets: [{
            data: sentimentData,
            backgroundColor: ['rgba(75, 192, 192, 1)', 'rgba(255, 206, 86, 1)', 'rgba(255, 99, 132, 1)'],
            borderColor: ['rgba(75, 192, 192, 1)', 'rgba(255, 206, 86, 1)', 'rgba(255, 99, 132, 1)'],
            borderWidth: 1
        }]
    },
    options: {
        responsive: true,
        plugins: {
            legend: {
                display: true,
                position: 'bottom',
                aligned: 'start',
            },
            tooltips: {
                callbacks: {
                    label: function (tooltipItem, data) {
                        return data.labels[tooltipItem.index]; // Show labels with percentages in tooltips
                    }
                }
            }
        }
    }
});
</script>
</body>
</html>
```

BASE HTML

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">
<title>{% block title %}{% endblock %}</title>
<!-- <link rel="stylesheet" type="text/css" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"/> -->
<link rel = "stylesheet" type = "text/css" href = "https://cdn.datatables.net/1.12.0/css/jquery.dataTables.min.css"/>
<link rel="stylesheet" type="text/css" href="{% static 'CSS/crud.css' %}">
<meta charset="UTF-8" name="viewport" content="width=device-width, initial-scale=1"/>
<!-- favicon <dev elgene -->
<link rel="icon" href="{% static 'elgene.png' %}" type="image/x-icon">
<link rel="shortcut icon" href="{% static 'favicon.ico' %}" type="image/x-icon">
<!-- Add these lines to your head section in base.html or the template <dev elgene> -->
  <script src="https://code.jquery.
```

```
com/jquery-
3.5.1.slim.min.js"></scri
pt>
  <script
src="https://cdn.jsdelivr.
net/npm/@popperjs/core
@2.5.3/dist/umd/popper
.min.js"></script>
  <script
src="https://stackpath.bo
otstrapcdn.com/bootstra
p/4.5.2/js/bootstrap.min.
js"></script>
</head>
<body>
<div class="main-
container">
 <div class="row">
   <div class="center-
text">
     <h1>Department
Management</h1>
  </div>
   {% block content
%}{% endblock %}
 </div>
</div>

<script
src="https://cdn.jsdelivr.
net/npm/jquery@3.6.0/d
ist/jquery.min.js"></scri
pt>
<script
src="https://cdn.jsdelivr.
net/npm/bootstrap@5.0.
2/dist/js/bootstrap.bundl
e.min.js"></script>
<script src =
"https://cdn.datatables.n
et/1.12.0/js/jquery.dataT
ables.min.js"></script>
<script>
$(document).ready(funct
ion() {
$('#bootstrapdatatable').
DataTable({
     "aLengthMenu":
[[10, 20, 30, 40, 50, -1],
[10, 20, 30, 40, 50,
"All"]],
      "iDisplayLength":
10
     }
   );
} );
</script>
</body>
</html>
```

EDIT HTML

```
{% extends
"basedepartment.html"
%}

{% block title %}Edit
Department{% endblock
title %}

{% block content %}
<div class="col-md-12-
edit">
   <form
method="POST"
class="post-form2"
action="{% url
'updatedepartment'
department.id %}">
      {% csrf_token %}
   <div
class="container">
      <div class="col-
12">
         <h1>Update
Details</h1>
      </div>
      <div class="mb-
3">
<label>Department
Name:</label>
         <input
type="text" class="form-
control" name="name"
id="id_name" required
maxlength="100"
value="{{
department.name }}" />
      </div>
      {% if messages %}
         <div class="mb-
3">
            {% for
message in messages %}
               <div
class="alert alert-
danger" role="alert"
style="color: red;">
                  {{
message }}
               </div>
            {% endfor %}
         </div>
      {% endif %}
      <div class="mb-3
button-container">
         <span
class="btn-add-new-2">
            <a href="{%
url 'showdepartment'
%}" style="font-weight:
normal; font-
family:'Times New
Roman', Times, serif;
font-size:
15px;">Back</a>
         </span>
         <button
type="submit"
class="btn btn-success
btn-lg" style="font-
weight: normal; font-
family:'Times New
Roman', Times, serif;
```

```
font-size:
15px;">Update</button
>
        </div>
    </div>
    </form>
{% endblock content %}
```

INDEX HTML

```
{%                extends
"basedepartment.html"
%}

{% block title %}Add
New Department{%
endblock title %}

{% block content %}
<div class="col-md-12-
add">
    <form
method="POST"
class="post-form2"
action="/departmentsdas
hboard/addnewdepartme
nt">
    {% csrf_token %}
    <div
class="container">
        <div class="col-
12">
        <h3>Enter
Department
Details</h3>
        </div>

        <div class="mb-
3">
            <label
for="fullName"
class="form-
label">Department
Name:</label>
            {{ form.name }}
            <br>
            {%          if
form.errors.name %}
                <span
class="error"
style="color: red;">{{
form.errors.name.0
}}</span>
            {% endif %}
        </div>
        <div class="mb-3
button-container">
            <span
class="btn-add-new-2">
                <a href="{%
url 'showdepartment'
%}" style="font-weight:
normal; font-
family:'Times New
Roman', Times, serif;
font-size:
15px;">Back</a>
            </span>
            <button
type="submit"
class="btn btn-success
btn-lg" style="font-
weight: normal; font-
family:'Times New
Roman', Times, serif;
font-size:
15px;">Submit</button
>
        </div>
    </div>
    </form>
</div>
{% endblock content %}
```

SHOW HTML

```
{%                extends
"basedepartment.html"
%}

{%        block        title
%}Manage
Departments{%
endblock title %}

{% block content %}
<div class="col-md-
12">
    <div class="button-
container">
        <span class="btn-
add-new">
            <a href="{% url
'AdminDashboard'
%}">HOME</a>
            <a href="{% url
'addnewdepartment'
%}">Add New
Department</a>
        </span>
    </div>
    <div class="table-
responsive">
        <table
id="bootstrapdatatable"
class="table table-
striped table-bordered"
width="100%">
            <thead>
                <th>ID</th>

<th>Name</th>

<th>Action</th>
            </thead>
            <tbody>
                {%          for
department in
departments %}
                    <tr>
                        <td>{{
department.id }}</td>
                        <td>{{
department.name
}}</td>
                        <td>
                            <div
style="display: flex;
```

```
flex-flow: row; align-items: center; justify-content: center; flex-wrap: wrap; width: 100%;">
                    <a href="{% url 'editdepartment' department.id %}" class="btn btn-primary">Edit</a>
                    <div style="margin: 0 2px 0 2px;"> </div>

<button class="btn btn-danger btn-confirm-action" data-toggle="modal" data-target="#deleteModal{{ department.id }}" data-url="{% url 'deletedepartment' department.id %}">Delete</button>
                    </div>
                  </td>
                </tr>
                <!-- Modal -->
                <div class="modal fade" id="deleteModal{{ department.id }}" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
                  <div class="modal-dialog" role="document">
                    <div class="modal-content">
                      <div class="modal-header">

<h5 class="modal-title" id="deleteModalLabel">Confirm Delete</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>
                      <div class="modal-body">

Are you sure you want to delete {{ department.name }}?

</div>
                      <div class="modal-footer">

<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>

<a href="{% url 'deletedepartment' department.id %}" id="confirmDeleteButton" class="btn btn-danger">Delete</a>

</div>
                    </div>
                  </div>
                <!-- End Modal -->
                {% endfor %}
              </tbody>
            </table>
          </div>
        </div>
{% endblock content %}
```

LOGIN CSS

Your cleaned-up CSS code without comments:

```css
@import url('https://fonts.googleapis.com/css?family=Raleway');

*{
  box-sizing: border-box;
}

.color{
  background: linear-gradient(to right, #e1e0ff, #d7fadd);
}

body{
  background: green;
  user-select: none;
  letter-spacing: 3px;
  color: #b9b9b9;
  overflow: hidden;
}

p{
  padding-bottom: 15px;
}

p:last-of-type{
  padding-bottom: 0;
```

```
}

.wrapper {
   position: absolute;
   left: 25%;
   top: 100px;
   width: 100%;
   height: 100vh;
   display: flex;
   flex-direction:
column;
   align-items: center;
   justify-content:
center;
}

.background-image {
   width: auto;
   height: 700px;
   margin: 25px 10px
10px 10px;
   position: absolute;
   z-index: -1;
   left: 100px;
}

.logo {
   width: 200px;
   position: absolute;
   top: 150px;
   left: 75%;
   transform:
translateX(-50%);
   z-index: 2;
}

.button {
   cursor: pointer;
   position: relative;
   z-index: 2;
   -webkit-appearance:
none;
   appearance: none;
   background: none;
   outline: none;
   border: none;
   text-transform:
uppercase;
   letter-spacing: 3px;
   border-radius: 2px;
   transition: all 0.5s
ease;
   font-weight: 600;
   font-size: 14px;
   display: flex;
   align-items: center;
   justify-content:
center;
}

.button-text,
.button-text p {
   display: inline-block;
   position: relative;
   z-index: 2;
   background: -webkit-
linear-gradient(to right,
#FFD700, #4cfa63);
   background: linear-
gradient(to right,
#FFD700, #4cfa63);
   -webkit-background-
clip: text;
   background-clip: text;
   color: transparent;
   font-size: 60px;
   text-align: center;
}

.button-text p {
   font-size: 40px;
   display: flex;
   justify-content:
center;
   align-items: center;
   color: white;
}

.button-text h1{
   padding-bottom:
20px;
   text-transform:
uppercase;
   letter-spacing: 3px;
   font-size: 50px;
   color: white;
}

.button:hover .button-
text p,
.button:hover .button-
text h1 {
   color: transparent;
}

.button-style2 {
   background-color:
#FFD700;
   color: #ffffff;
   border: none;
   padding: 10px 20px;
   font-size: 16px;
   cursor: pointer;
   border: 1px solid
rgba(0,0,0,1);
   border-radius: 5px;
   transition:
background-color 0.3s;
   text-decoration: none;
   margin-top: 200px;
}

.button-style2:hover {
   background-color:
#ffffff;
   color: #FFD700;
}

.button-style2 p {
   margin: 0;
}

.button-circle{
   position: absolute;
   left: 0;
   top: 0;
   height: 6px;
```

```
    width: 6px;
    border-radius: 50%;
    box-shadow: 0 0 10px
#FFD700, 0 0 20px
#FFD700, 0 0 30px
#FFD700;
    transition: all 0.5s
ease;
}

.button-circle2{
    top: auto;
    bottom: 0;
}

.button-circle3{
    left: auto;
    right: 0;
    box-shadow: 0 0 10px
#4cfa63, 0 0 20px
#4cfa63, 0 0 30px
#4cfa63;
}

.button-circle4{
    left: auto;
    top: auto;
    bottom: 0;
    right: 0;
    box-shadow: 0 0 10px
#4cfa63, 0 0 20px
#4cfa63, 0 0 30px
#4cfa63;
}

.popup{
    opacity: 0;
    visibility: hidden;
    height: 400px;
    width: 400px;
    flex-shrink: 0;
    border-radius: 3px;
    position: relative;
    z-index: 3;
    display: flex;
    align-items: center;
    justify-content:
center;
    transition: all 0.2s
ease;
    right: 400px;
    bottom: 100px;
}

.popup-inside{
    position: absolute;
    left: 0;
    top: 0;
    height: 100%;
    width: 100%;
    overflow: hidden;
    border-radius: 50%;
    box-shadow: 0 0 0
black;
    transition:
        box-shadow 0.5s
ease 0.7s,
        border-radius
0.35s ease 0.7s;
}

.backgrounds{
    position: absolute;
    left: 0;
    top: 0;
    height: 100%;
    width: 100%;
    overflow: hidden;
}

.background{
    --offset: 0;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    background: linear-
gradient(to right,
#ffea00, #4cfa63);
    transform: scale(0);
    transition: all 0.5s ease
0s;
    border-radius: 50%;
}

.background2{
    --offset: 10%;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    background: linear-
gradient(to right,
#ffdd00, #69fa7f);
    transform: scale(0);
    transition: all 0.5s ease
0.1s;
}

.background3{
    --offset: 20%;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    background: linear-
gradient(to right,
#ffd000, #85fa99);
    z-index: 2;
    transition: all 0.5s ease
0.2s;
    border-radius: 20px;
}

.background4{
    --offset: 30%;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    background: linear-
gradient(to right,
#ffc300, #a7fab7);
```

```
    z-index: 3;
    transition: all 0.5s ease
0.3s;
    border-radius

: 20px;
}

.background5{
    --offset: 40%;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    background: linear-
gradient(to right,
#ffb700, #c3fad1);
    z-index: 4;
    transition: all 0.5s ease
0.4s;
    border-radius: 20px;
}

.background6{
    --offset: 40%;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    background: white;
    z-index: 5;
    transition: all 0.8s ease
0.4s;
    border-radius: 20px;
}

.content{
    --offset: 0;
    position: absolute;
    left: var(--offset);
    right: var(--offset);
    bottom: var(--offset);
    top: var(--offset);
    display: flex;
    align-items: center;
    justify-content:
center;
    opacity: 0;
    transition: all 0.35s
ease 0.75s;
    z-index: 10;
    border-radius: 20px;
}

body.active .content{
    opacity: 1;
    transform: none;
}

body.active .popup{
    opacity: 1;
    visibility: visible;
    text-align: center;
    width: 300px;
    height: 400px;
}

body.active .popup-
inside{
    border-radius: 30px;
}

body.active
.background{
    transform: scale(1);
}

body.active
.background6{
    transform: scale(8);
}

.button-style {
    background-color:
#FFD700;
    color: #ffffff;
    border: none;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    border-radius: 5px;
    transition:
background-color 0.3s;
    text-decoration: none;
}

.button-style:hover {
    background-color:
green;
}

.content label {
    display: block;
    text-align: center;
    margin-top: 10px;
}

.content
input[type="text"],
.content
input[type="password"]
{
    width: 100%;
    padding: 10px;
    margin: 5px 0;
    border: 1px solid
#ccc;
    border-radius: 3px;
    transition: border 0.3s;
}

.content
input[type="text"]:hover
,
.content
input[type="password"]:
hover {
    border: 1px solid
#555;
    background-color:
#f2f2f2;
}

.content button {
    background-color:
#FFD700;
```

```
    color: #ffffff;
    border: none;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    border-radius: 5px;
    transition:
background-color 0.3s;
    text-decoration: none;
}

.content button:hover {
    background-color:
green;
}

@media only screen and
(max-width: 600px) {
    .logo {
        width: 100px;
        position: absolute;
        top: 20%;
        left: 50%;
        transform:
translate(-50%, -50%);
        z-index: -1;
    }

    .background-image {
        width: 250px;
        height: 400px;
        position: absolute;
        top: 70%;
        left: 47%;
        transform:
translate(-50%, -50%);
        z-index: -1;
    }

    .button-text p {
        font-size: 14px;
        font-weight: bold;
    }

    .button-text h1 {
        font-size: 30px;
        padding-bottom:
10px;
    }

    .popup {
        max-height: 100%;
        max-width: 100%;
        bottom: 70px;
        left: 0;
        right: 0;
    }

    .popup-inside {
        height: 130%;
        overflow: hidden;
        margin: 10px;
        width: 95%;
    }

    .backgrounds {
        height: 100%;
    }

    .button {
        width: 80%;
        height: 250px;
        overflow: hidden;
    }

    .button-text p {
        font-size: 14px;
    }

    .button-text h1 {
        font-size: 30px;
        padding-bottom:
10px;
    }
}
```

ADMIN DASHBOARD CSS

Here is the cleaned-up CSS code without comments:

```
```css
body {
  background-color:
aqua;
  margin: 20px;
  background-color:
#f6fdfb;
}

.main-container {
  background-color:
green;
  margin: 20px;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 10px
20px rgba(0, 0, 0, 0.2);
  text-align: center;
}

.center-text {
  color: white;
  font-weight: bold;
  position: relative;
  margin-bottom: 70px;
}

.center-text h1 {
  font-size: 60px;
  color: gold;
  font-weight: bold;
  text-shadow: 2px 2px
rgb(165, 140, 0);
}

.account-icon {
  position: absolute;
  top: 30px;
  right: 60px;
  cursor: pointer;
  font-size: 25px;
  z-index: 1;
  color: white;
}
```
```

```
.dropdown-content {
  display: none;
  position: absolute;
  background-color:
#f9f9f9;
  min-width: 180px;
  box-shadow: 0px 8px
16px 0px rgba(0, 0, 0,
0.10);
  z-index: 1;
  right: 0;
  border-radius: 10px;
}

#account-
dropdown:hover
.dropdown-content {
  display: block;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-content
a:hover {
  background-color:
#f1f1f1;
  border-radius: 10px;
}

.tools {
  background-color:
#f9f9f9;
  box-shadow: 0 10px
20px rgba(0, 0, 0, 0.10);
  border-radius: 10px;
  padding: 10px;
  height: auto;
  width: 100%;
  position: relative;
  display: flex;
  justify-content: space-
between;
  align-items: center;
}

.select-staff-container {
  text-align: center;
  font-size: 20px;
  padding: 5px 10px;
  font-weight: bold;
  display: inline;
}

.date-filter {
  text-align: center;
  font-size: 20px;
  padding: 5px 10px;
  font-weight: bold;
  display: inline;
}

button {
  background-color:
green;
  color: #fff;
  font-size: 1.2em;
  padding: 10px 20px;
  border-radius: 10px;
  box-shadow: none;
}

button:hover {
  background-color:
rgb(90, 199, 90);
}

.left-container {
  width: calc(50% -
20px);
  background-color:
#f9f9f9;
  box-shadow: 0 10px
20px rgba(0, 0, 0, 0.10);
  border-radius: 10px;
  padding: 20px;
  margin: 20px 20px
20px 0;
  float: left;
  height: 400px;
}

.receipt-details {
  margin-bottom: 10px;
  font-weight: bold;
}

.item {
  display: flex;
  justify-content: space-
between;
  margin-bottom: 5px;
}

.item span {
  padding-left: 20px;
}

.total {
  display: flex;
  justify-content: space-
between;
  font-weight: bold;
}

.pie-chart-container {
  background-color:
#f9f9f9;
  box-shadow: 0 10px
20px rgba(0, 0, 0, 0.10);
  border-radius: 10px;
  padding: 10px;
  margin: 20px 0 20px
20px;
  float: left;
  height: 400px;
  width: calc(50% -
20px);
  position: relative;
}

#sentimentPieChart {
```

```
  width: 100%;
  height: 100%;
}

.chart-title {
  position: absolute;
  top: 50%;
  right: 3%;
  transform: translate(0, -
50%);
  text-align: center;
  font-size: 25px;
  padding: 5px 10px;
  font-weight: bold;
}

.departmentcount-
container {
  background-color:
#f9f9f9;
  box-shadow:  0  10px
20px rgba(0, 0, 0, 0.10);
  border-radius: 10px;
  padding: 10px;
  margin:  10px  10px
30px 0;
  float: left;
  height: auto;
  width: 100%;
  position: relative;
}

#departmentChart {
  display: block;
  box-sizing: border-box;
  max-width: 100%;
  max-height: 400px;
  height: auto;
}

.departmentcount-
container h2 {
  font-size: 1.5em;
  color: #333;
  margin-bottom: 10px;
}

.table-container {
  background-color: #fff;
  box-shadow:  0  20px
40px rgba(0, 0, 0, 0.10);
  padding: 15px;
  max-width: 100%;
  margin: auto;
  border-radius: 10px;
  clear: both;
  position: relative;
  overflow-x: auto;
}

.table-wrapper {
  max-height: 600px;
  overflow-y: auto;
  border-radius: 8px;
  margin-top: 20px;
  clear: both;
  position: relative;
  width: 100%;
}

.fl-table {
  border-radius: 5px;
  font-size: 12px;
  font-weight: normal;
  border: none;
  border-collapse:
collapse;
  background-color:
white;
  table-layout: auto;
}

.fl-table td,
.fl-table th {
  text-align: center;
  white-space: normal;
  word-wrap:  break-
word;
  border-right: 1px solid
#f8f8f8;
}

.fl-table td {
  border-right: 1px solid
#f8f8f8;
  font-size: 12px;
}

.fl-table  thead  th:nth-
child(odd) {
  color: #ffffff;
  background: green;
}

.fl-table  tr:nth-
child(even) {
  background: #d6d4d4;
}

.fl-table thead {
  position: sticky;
  top: 0;
  background:  rgb(90,
199, 90);
  color: #ffffff;
  z-index: 2;
}

.dataTables_wrapper
.dataTables_paginate
.paginate_button:hover {
  color: white !important;
  border:  1px  solid
#006400;
  background-color:
#006400;
  background:  -webkit-
gradient(
    linear,
    left top,
    left bottom

,
    color-stop(0%,
#585858),
    color-stop(100%,
#111)
  );
```

```
  background: -webkit-linear-gradient(top, #006400 0%, #003300 100%);
  background: -moz-linear-gradient(top, #006400 0%, #003300 100%);
  background: -ms-linear-gradient(top, #006400 0%, #003300 100%);
  background: -o-linear-gradient(top, #006400 0%, #003300 100%);
  background: linear-gradient(to bottom, #006400 0%, #003300 100%);
}

.modal,
.modal2 {
  display: none;
  position: fixed;
  z-index: 3;
  left: 0;
  top: 0;
  width: 100%;
  height: 100%;
  overflow: hidden;
  background-color: rgba(0, 0, 0, 0.4);
  border-radius: 10px;
}

.modal-content {
  background-color: #fefefe;
  margin: 15% auto;
  padding: 20px;
  border: 1px solid #888;
  width: 50%;
  position: relative;
  border-radius: 10px;
}

.close {
  color: #aaaaaa;
  position: absolute;
  top: 0;
  right: 15px;
  font-size: 28px;
  font-weight: bold;
  cursor: pointer;
}

.close:hover,
.close:focus {
  color: red;
  text-decoration: none;
  cursor: pointer;
}

.modal button {
  margin: 10px;
}

@media (max-width: 767px) {
  .main-container {
    margin: 10px;
  }
  .left-container,
  .pie-chart-container,
  .table-container {
    width: 100%;
    margin: 0;
  }
  .date-filter label {
    margin: 0 5px;
  }

  .date-filter input {
    width: auto;
  }
  .date-filter button {
    margin: 0 5px;
    padding: 5px 10px;
  }
  .fl-table {
    font-size: 14px;
  }
  .chart-title {
    top: 350px;
  }
}
```

DEPARTMENT DASHBOARD CSS

```
body {
background-color: aqua;
margin: 20px;
background-color: #f6fdfb;
}
.main-container {
background-color: green;
margin: 20px;
padding: 20px;
border-radius: 10px;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);
text-align: center;
}
.center-text{
color: white;
font-weight: bold;
position: relative;
margin-bottom: 70px;
}
.center-text h1{
font-size: 60px;
color: gold;
font-weight: bold;
text-shadow: 2px 2px rgb(165, 140, 0);
}


/*account logo */
.account-icon {
position: absolute;
top: 30px;
right: 60px;
cursor: pointer;
```

```
font-size: 25px; /* Adjust the font size as needed */
z-index: 1;
color: white;
}
.dropdown-content {
display: none;
position: absolute;
background-color: #f9f9f9;
min-width: 180px;
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.10);
z-index: 1;
right: 0;
border-radius: 10px;
}
#account-dropdown:hover .dropdown-content {
display: block;
}
.dropdown-content a {
color: black;
padding: 12px 16px;
text-decoration: none;
display: block;
}
.dropdown-content a:hover {
background-color: #f1f1f1;
border-radius: 10px;
}
/* end of account logo*/


/*staff/date/filter*/
.tools {
background-color: #f9f9f9;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.10);
border-radius: 10px;
padding: 10px;
height: auto;
width: 100%;
position: relative;
display: flex;
justify-content: space-between;
align-items: center;
}
.select-staff-container {
text-align: center;
font-size: 20px;
padding: 5px 10px;
font-weight: bold;
display: inline;
}
.date-filter {
text-align: center;
font-size: 20px;
padding: 5px 10px;
font-weight: bold;
display: inline;
}
button {
background-color: green;
color: #fff;
font-size: 1.2em;
padding: 10px 20px;
border-radius: 10px;
box-shadow: none;
}
button:hover {
background-color:rgb(90, 199, 90);
}
/*staff/date/filter*/


/* AREAS OF CONCERN */
.left-container {
width: calc(50% - 20px);
background-color: #f9f9f9;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.10);
border-radius: 10px;
padding: 20px;
margin: 20px 20px 20px 0;
float: left;
height: 400px;
}
.receipt-details {
margin-bottom: 10px;
font-weight: bold;
}
.item {
display: flex;
justify-content: space-between;
margin-bottom: 5px;
}
.item span {

    padding-left: 20px;
}
.total {
display: flex;
justify-content: space-between;
font-weight: bold;
}
/*areas of concern*/


/* pie chart */
.pie-chart-container {
background-color: #f9f9f9;
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.10);
border-radius: 10px;
padding: 10px;
margin: 20px 0 20px 20px;
float: left;
height: 400px;
width: calc(50% - 20px);
position: relative;
```

```
}
#sentimentPieChart {
width: 100%;
height: 100%;
}
element.style {
display: block;
box-sizing: border-box;
height: 380px;
width: 500px;
}
.chart-title {
position: absolute;
top: 50%;
right: 3%;
transform: translate(0, -50%);
text-align: center;
font-size: 25px;
padding: 5px 10px;
font-weight: bold;
}
/* pie chart */


/*history*/
.table-container {
background-color: #fff;
box-shadow: 0 20px 40px rgba(0, 0, 0, 0.10);
padding: 15px;
max-width: 100%;
margin: auto;
border-radius: 10px;
clear: both;
position: relative;
overflow-x: auto;
}
.table-wrapper {
max-height: 600px;
overflow-y: auto;
border-radius: 8px;
margin-top: 20px;
clear: both;
position: relative;
width: 100%;
}
.fl-table {
border-radius: 5px;
font-size: 12px;
font-weight: normal;
border: none;
border-collapse: collapse;
background-color: white;
table-layout: auto;
}
.fl-table td,
.fl-table th {
text-align: center;
white-space: normal;
word-wrap: break-word;
border-right: 1px solid #f8f8f8;
}
.fl-table td {
border-right: 1px solid #f8f8f8;
font-size: 12px;
}
.fl-table thead th:nth-child(odd) {
color: #ffffff;
background: green;
}
.fl-table tr:nth-child(even) {
background: #d6d4d4;
}
.fl-table thead {
position: sticky;
top: 0;
background: rgb(90, 199, 90);
color: #ffffff;
z-index: 2;
}
.dataTables_wrapper .dataTables_paginate .paginate_button:hover {
color: white !important;
border: 1px solid #006400; /* Dark green */
background-color: #006400; /* Dark green */
background: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #585858), color-stop(100%, #111));
background: -webkit-linear-gradient(top, #006400 0%, #003300 100%);
background: -moz-linear-gradient(top, #006400 0%, #003300 100%);
background: -ms-linear-gradient(top, #006400 0%, #003300 100%);
background: -o-linear-gradient(top, #006400 0%, #003300 100%);
background: linear-gradient(to bottom, #006400 0%, #003300 100%);
}
/*history*/

/* modal */
.modal, .modal2 {
display: none;
position: fixed;
z-index: 3;
left: 0;
top: 0;
width: 100%;
height: 100%;
overflow: hidden;
```

```
background-color:
rgba(0,0,0, 0.4);
border-radius: 10px;
}
.modal-content {
background-color:
#fefefe;
margin: 15% auto;
padding: 20px;
border: 1px solid #888;
width: 50%;
position: relative;
border-radius: 10px;
}
.close {
color: #aaaaaa;
position: absolute;
top: 0;
right: 15px;
font-size: 28px;
font-weight: bold;
cursor: pointer;
}
.close:hover,
.close:focus {
color: red;
text-decoration: none;
cursor: pointer;
}
.modal button{
margin: 10px;
}
/* modal */

@media (max-width:
767px) {
.main-container {
   margin: 10px;
}
.left-container, .pie-
chart-container, .table-
container {
   width: 100%;
   margin: 0;
}
.date-filter label {
   margin: 0 5px;
}

.date-filter input {
   width: auto;
}
.date-filter button {
   margin: 0 5px;
   padding: 5px 10px;
}
.fl-table {
   font-size: 14px;
}
.chart-title{
   top: 350px;
}
}
```

CRUD CSS

Here is the cleaned-up CSS code without comments:

```css
body {
   background-color:
#f6fdfb;
}

.main-container {
   background-color:
green;
   margin: 20px;
   padding: 10px;
   border-radius: 10px;
   box-shadow: 0 10px
20px rgba(0, 0, 0, 0.2);
   text-align: center;
}

.center-text h1 {
   text-align: center;
   font-size: 60px;
   color: gold;
   font-weight: bold;
   margin: 20px 20px
65px 20px;
   text-shadow: 2px 2px
rgb(165, 140, 0);
}

.col-md-12 {
   max-width: 95%;
   margin: 0 auto;
   background-color:
white;
   padding: 1px 20px
20px 20px;
   border-radius: 10px;
}

.button-container {
   display: flex;
   justify-content: flex-
start;
   align-items: center;
   margin-top: 0px;
   margin-bottom: 10px;
}

.btn-add-new a {
   display: inline-block;
   padding: 10px 20px;
   margin: 0 10px 20px
10px;
   text-decoration: none;
   background-color:
green;
   color: #fff;
   border: 1px solid
green;
   border-radius: 5px;
   transition:
background-color 0.3s,
color 0.3s;
}

.btn-add-new a:hover {
   background-color:
#27ae60;
   color: #fff;
```

```
}

.btn-primary {
   display: inline-block;
   padding: 10px 20px;
   margin: 10px 0;
   text-align: center;
   text-decoration: none;
   background-color:
green;
   color: #fff;
   border: none;
   border-radius: 5px;
   transition:
background-color 0.2s;
}

.btn-primary:hover {
   background-color:
rgb(90, 199, 90);
}

.btn-danger {
   display: inline-block;
   padding: 10px 20px;
   margin: 10px 0;
   text-align: center;
   text-decoration: none;
   background-color:
red;
   color: #fff;
   border: none;
   border-radius: 5px;
   transition:
background-color 0.2s;
}

.btn-danger:hover {
   background-color:
#ff3333;
}

#bootstrapdatatable {
   width: 80%;
   margin: 0 auto;
}

#bootstrapdatatable
thead th:nth-child(odd) {
   background-color:
green;
   color: #fff;
}

#bootstrapdatatable
thead   th:nth-child(even)
{
   background-color:
rgb(90, 199, 90);
   color: #fff;
}

#bootstrapdatatable
tbody tr:nth-child(odd) {
   background-color:
#f7f7f7;
}

#bootstrapdatatable
tbody   tr:nth-child(even)
{
   background-color:
#fff;
}

.post-form {
   background-color:
#f5f5f5;
   padding: 50px;
   border:     1px     solid
#ddd;
   border-radius: 10px;
   box-shadow:  0  0  5px
rgba(0, 0, 0, 0.2);
   max-width: 15%;
   margin: 0 auto;
   position: relative;
   font-weight: bold;
   text-align: center;
   margin-bottom: 10vh;
}

.post-form2 {
   background-color:
#f5f5f5;
   padding: 50px;
   border:     1px     solid
#ddd;
   border-radius: 10px;
   box-shadow:  0  0  5px
rgba(0, 0, 0, 0.2);
   max-width: 15%;
   margin: 0 auto;
   position: relative;
   font-weight: bold;
   text-align: center;
   margin-bottom: 29vh;
}

.col-12,
.mb-3 {
   position: relative;
   align-items: center;
   margin-bottom: 30px;
}

.post-form label {
   display: block;
   font-weight: bold;
}

.post-form
input[type="text"],
.post-form textarea,
.post-form select {
   max-width: 100%;
   padding: 10px;
   margin-bottom: 10px;
   border:     1px     solid
#ccc;
   border-radius: 3px;
}

.post-form button {
   background-color:
green;
   color: #fff;
   padding: 10px 20px;
```

```
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

.post-form2 button {
    background-color:
green;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

.btn-add-new-2 a {
    text-decoration: none;
    background-color:
green;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
    font-weight: normal;
}

.button-container {
    display: flex;
    justify-content: space-
between;
    align-items: center;
    margin-top: 50px;
    margin-bottom: 0;
}

.dataTables_wrapper
.dataTables_paginate
.paginate_button:hover {
    color:          white
!important;
    border:    1px    solid
#006400;
    background-color:
#006400;
    background:  -webkit-
gradient(
        linear,
        left top,
        left bottom,
        color-stop(0%,
#585858),
        color-stop(100%,
#111)
    );
    background:  -webkit-
linear-gradient(top,
#006400 0%, #003300
100%);
    background:     -moz-
linear-gradient(top,
#006400 0%, #003300
100%);
    background:     -ms-
linear-gradient(top,
#006400 0%, #003300
100%);
    background: -o-linear-
gradient(top, #006400
0%, #003300 100%);
    background:    linear-
gradient(to bottom,
#006400 0%, #003300
100%);
}

.modal {
    position: fixed;
    top:

0;
    right: 0;
    bottom: 0;
    left: 0;
    z-index: 1050;
    overflow: hidden;
    outline: 0;
    display: none;
    background-color:
rgba(0, 0, 0, 0.5);
}

.modal-dialog {
    position: relative;
    width: 80%;
    max-width: none;
    top: 35%;
    left: 30%;
}

.modal-content {
    position: relative;
    display: flex;
    flex-direction:
column;
    width: 50%;
    pointer-events: auto;
    background-color:
#fff;
    background-clip:
padding-box;
    border:    1px    solid
rgba(0, 0, 0, 0.2);
    border-radius: 0.3rem;
    outline: 0;
}

.modal-header {
    display: flex;
    align-items: flex-start;
    justify-content: space-
between;
    padding: 15px;
    border-bottom:    1px
solid #e9ecef;
    border-top-left-radius:
0.3rem;
    border-top-right-
radius: 0.3rem;
}

.modal-title {
    margin-bottom: 0;
    line-height: 1.5;
}

.modal-body {
```

```
    position: relative;
    flex: 1 1 auto;
    padding: 15px;
}

.modal-footer {
    display: flex;
    align-items: center;
    justify-content: flex-end;
    padding: 15px;
    border-top: 1px solid #e9ecef;
    border-bottom-right-radius: 0.3rem;
    border-bottom-left-radius: 0.3rem;
}

.btn-secondary {
    display: inline-block;
    padding: 10px 20px;
    margin: 10px 20px;
    text-align: center;
    text-decoration: none;
    background-color: green;
    color: #fff;
    border: none;
    border-radius: 5px;
    transition: background-color 0.2s;
}

.btn-danger {
    color: #fff;
    background-color: #dc3545;
    border-color: #dc3545;
}
```

FORMS.PY

```
from django import forms
from visithoughtsForm.models import Department, Staff
class StaffForm(forms.ModelForm):
        name = forms.CharField(

widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter Staff Full Name'}),
        )
        department = forms.ModelChoiceField(

queryset=Department.objects.all(),
            label='Select Department',

widget=forms.Select(attrs={'class': 'form-select'}),
        )
        class Meta:
            model = Staff
            fields = '__all__'

class DepartmentForm(forms.ModelForm):
    class Meta:
        model = Department
        fields = ['name']

    def clean_name(self):
        name = self.cleaned_data['name']
        existing_department = Department.objects.filter(name=name)

        if existing_department.exists():
            raise forms.ValidationError('A department with this name already exists.')

        return name
```

URLS.PY

```
from django.urls import path
from . import views

urlpatterns = [
    # <dev elgene>
    path('login/', views.custom_login, name='login'),
    path('logout/', views.custom_logout, name='logout'),

path('DepartmentDashboard/', views.DepartmentDashboard, name='DepartmentDashboard'),

path('AdminDashboard/', views.AdminDashboard, name='AdminDashboard'),

path('feedback/ajax/load-staffs/', views.load_staffs,
```

```
name='ajax_load_staffs'), # AJAX

path('change_username/', views.change_username, name='change_username'),

path('change_password/', views.change_password, name='change_password'),
    # <dev elgene>
    path('show', views.index, name='show'),

path('addnew',views.addnew, name='addnew'),
    path('edit/<int:id>', views.edit, name='edit'),
    path('update/<int:id>', views.update, name='update'),
    path('delete/<int:id>', views.destroy, name='delete'),
    # <dev elgene>

path('showdepartment', views.indexdepartment, name='showdepartment'),

path('addnewdepartment',views.addnewdepartment, name='addnewdepartment'),

path('editdepartment/<int:id>', views.editdepartment, name='editdepartment'),

path('updatedepartment/<int:id>', views.updatedepartment, name='updatedepartment'),

path('deletedepartment/<int:id>', views.destroydepartment, name='deletedepartment'),
]
```

VIEWS.PY

```
from django.shortcuts import render, redirect
from visithoughtsForm.models import Feedback, Staff, Department
from collections import Counter
from datetime import datetime
from django.contrib.auth import authenticate, login
from django.contrib.auth.decorators import login_required
from django.contrib.auth import logout
from django.http import JsonResponse
from django.contrib.auth import update_session_auth_hash
from django.contrib.auth.forms import PasswordChangeForm
from django.contrib.auth.models import User
from .forms import StaffForm, DepartmentForm
from django.contrib import messages
from django.db.models import Count
from django.shortcuts import get_object_or_404

@login_required
def addnewdepartment(request):
    if request.method == "POST":
        form = DepartmentForm(request.POST)
        if form.is_valid():
            try:
                form.save()

messages.success(request, 'Department added successfully.')
                return redirect('showdepartment')
            except Exception as e:
                # Log the error for debugging purposes
                print(f"An error occurred: {e}")
                # Display an error message to the user

messages.error(request, 'An error occurred while adding the department. Please try again.')
```

```
    else:
        form = DepartmentForm()

    return render(request, 'indexdepartment.html', {'form': form})
# <dev elgene>
@login_required
def indexdepartment(request):
    departments = Department.objects.all().order_by('-id')
    return render(request, "showdepartment.html", {'departments': departments})
# <dev elgene>
@login_required
def editdepartment(request, id):
    department = Department.objects.get(id=id)
    return render(request, 'editdepartment.html', {'department': department})
# <dev elgene>
@login_required
def updatedepartment(request, id):
    department = get_object_or_404(Department, id=id)

    if request.method == "POST":
        form = DepartmentForm(request.POST, instance=department)
        if form.is_valid():
            new_name = form.cleaned_data['name']

            # Check if the new name is the same as the existing names

            existing_departments = Department.objects.exclude(id=id).filter(name=new_name)
            if existing_departments.exists():

                messages.error(request, 'Department with this name already exists. Please choose a different name.')
            else:
                form.save()

                messages.success(request, 'Department updated successfully.')
                return redirect('showdepartment')
        else:

            messages.error(request, 'Department with this name already exists. Please choose a different name.')

    else:
        form = DepartmentForm(instance=department)

    return render(request, 'editdepartment.html', {'department': department, 'form': form})
# <dev elgene>
@login_required
def destroydepartment(request, id):
    department = Department.objects.get(id=id)
    department.delete()
    return redirect('showdepartment')
# <dev elgene>
@login_required
def addnew(request):
    if request.method == "POST":
        form = StaffForm(request.POST)
        if form.is_valid():
            try:
                form.save()
                return redirect('show')
            except Exception as e:
                # Handle any exceptions here, for example, logging the error
                print(f"An error occurred: {e}")
                # You might want to display an error message to the user
    else:
        form = StaffForm()

    return render(request, 'index.html', {'form': form})
# <dev elgene>
```

```
@login_required
def index(request):
    staffs = Staff.objects.all()
    for staff in staffs:
        department = Department.objects.get(id=staff.department_id)

        staff.department_name = department.name  # Add department_name attribute to staff
    return render(request, "show.html", {'staffs': staffs})
# <dev elgene>
@login_required
def edit(request, id):
    department_names = Department.objects.all()
    staff = Staff.objects.get(id=id)
    staff_department = staff.department
    return render(request, 'edit.html', {'employee': staff, 'department_names': department_names, 'staff_department': staff_department})
# <dev elgene>
@login_required
def update(request, id):
    staff = Staff.objects.get(id=id)
    if request.method == "POST":
        form = StaffForm(request.POST, instance=staff)
        if form.is_valid():
            form.save()
            return redirect('show')
    else:
        form = StaffForm(instance=staff)

    department_names = Department.objects.all()
    return render(request, 'edit.html', {'staff': staff, 'form': form, 'department_names': department_names})
# <dev elgene>
@login_required
def destroy(request, id):
    employee = Staff.objects.get(id=id)
    employee.delete()
    return redirect('show')
# <dev elgene>
def custom_login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)

        if user is not None:
            first_name = user.first_name
            if first_name =='administrator':
                login(request, user)
                return redirect('AdminDashboard')
            elif first_name == 'Department Of Computer Studies':

                request.session['gene'] = 1
                login(request, user)
                return redirect('DepartmentDashboard')
            elif first_name == 'Department of Arts and Sciences':

                request.session['gene'] = 2
                login(request, user)
                return redirect('DepartmentDashboard')
            elif first_name == 'Department of Teacher Education':

                request.session['gene'] = 3
                login(request, user)
                return redirect('DepartmentDashboard')
            elif first_name == 'Department of Management Studies':

                request.session['gene'] = 4
                login(request, user)
                return redirect('DepartmentDashboard')
```

```
        elif first_name == 'Department of Engineering':
            request.session['gene'] = 5
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Department of Industrial Technology':
            request.session['gene'] = 6
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Library':
            request.session['gene'] = 7
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'MIS':
            request.session['gene'] = 8
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'OCA':
            request.session['gene'] = 9
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Human Resource':
            request.session['gene'] = 10
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Cashier':
            request.session['gene'] = 11
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Accounting':
            request.session['gene'] = 12
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Registrar':
            request.session['gene'] = 13
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'ODI':
            request.session['gene'] = 14
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Accreditation':
            request.session['gene'] = 15
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'GAD':
            request.session['gene'] = 16
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Supply':
            request.session['gene'] = 17
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'PPSS':
```

```
request.session['gene'] = 18
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Clinic':
            request.session['gene'] = 19
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'EBA':
            request.session['gene'] = 20
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Research and Extension':
            request.session['gene'] = 21
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'OJT':
            request.session['gene'] = 22
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Planning':
            request.session['gene'] = 23
            login(request, user)
            return redirect('DepartmentDashboard')
        elif first_name == 'Tesda':
            request.session['gene'] = 24
            login(request, user)
            return redirect('DepartmentDashboard')
        # Add more departments and their respective dashboard views as needed
    else:
        # Invalid credentials, show error message
        return render(request, 'login.html', {'error_message': 'Invalid credentials'})
    return render(request, 'login.html')
# <dev elgene>
def custom_logout(request):
    logout(request)
    return redirect('login')
# Redirect to the login page after logging out
# <dev elgene>
@login_required
def AdminDashboard(request):
    department_names = Department.objects.all()
    staff_members = Staff.objects.all()
    feedback_records = Feedback.objects.all()

    # You can access the data here
    selected_department_id = request.POST.get('department')
    selected_staff_id = request.POST.get('attendingStaff')
    start_date = request.POST.get('start-date')
    end_date = request.POST.get('end-date')

    if start_date and end_date:
        # Parse the start and end dates as datetime objects
        start_date = datetime.strptime(start_date, '%Y-%m-%d')
        end_date = datetime.strptime(end_date, '%Y-%m-%d')

        # Filter feedback records by date range
        feedback_records = feedback_records.filter(dateOfVisit__gte=start_date,
```

```
dateOfVisit__lte=end_date)

    if selected_department_id and selected_department_id != "":
        department = Department.objects.get(id=selected_department_id)
        feedback_records = feedback_records.filter(department=department)

        # Get department name
        selected_department_name = department
    else:
        selected_department_name = "Over All Evaluation"

    if selected_staff_id and selected_staff_id != "":
        staff = Staff.objects.get(id=selected_staff_id)
        feedback_records = feedback_records.filter(staff=staff)

        # Get staff name
        selected_staff_name = staff
    else:
        selected_staff_name = "None"

    # Initialize variables for calculating averages <dev elgene>
    total_records = 0
    total_courtesy = 0
    total_quality = 0
    total_timeless = 0
    total_efficiency = 0
    total_cleanliness = 0
    total_comfort = 0
    sentiments = []

    for feedback in feedback_records:
        total_records += 1
        total_courtesy += (feedback.courtesy) * 20 # Convert to a scale of 0 to 100 <dev elgene>
        total_quality += (feedback.quality) * 20
        total_timeless += (feedback.timeless) * 20
        total_efficiency += (feedback.efficiency) * 20
        total_cleanliness += (feedback.cleanliness) * 20
        total_comfort += (feedback.comfort ) * 20
        sentiments.append(feedback.sentiment)

    # Calculate average values for areas of concern <dev elgene>
    average_courtesy = total_courtesy / total_records if total_records > 0 else 0
    average_quality = total_quality / total_records if total_records > 0 else 0
    average_timeless = total_timeless / total_records if total_records > 0 else 0
    average_efficiency = total_efficiency / total_records if total_records > 0 else 0
    average_cleanliness = total_cleanliness / total_records if total_records > 0 else 0
    average_comfort = total_comfort / total_records if total_records > 0 else 0

    # Calculate the total average <dev elgene>
    total_average = (average_courtesy + average_quality + average_timeless + average_efficiency + average_cleanliness + average_comfort) / 6

    # Count sentiments
    sentiment_counts = dict(Counter(sentiments))
    total_sentiments = sum(sentiment_counts.values())

    selected_staff_id = request.POST.get('attendingStaff')
    start_date = request.POST.get('start-date')
    end_date = request.POST.get('end-date')
```

```
    # Count records for each department

records_per_department = Feedback.objects.values('department').annotate(count=Count('id'))

    # Extract data for plotting

department_names_for_plot = [item['department'] for item in records_per_department]

record_counts_for_plot = [item['count'] for item in records_per_department]

    # Count sentiments for each department

sentiment_counts_per_department = Feedback.objects.values('department', 'sentiment').annotate(count=Count('id'))

    # Print the raw data for debugging
    print("Raw Sentiment Data:", sentiment_counts_per_department)

    # Initialize dictionaries to store sentiment counts for each department

positive_sentiments_per_department = {}

negative_sentiments_per_department = {}

neutral_sentiments_per_department = {}

    for item in sentiment_counts_per_department:
        department = item['department']
        sentiment = item['sentiment'].strip() # Add strip() to remove leading/trailing spaces
        count = item['count']


print("Department:", department, "Sentiment:", sentiment, "Count:", count)  # Print for debugging

        if sentiment == 'Positive Sentiment':

positive_sentiments_per_department[department] = count
        elif sentiment == 'Negative Sentiment':

negative_sentiments_per_department[department] = count
        elif sentiment == 'Neutral Sentiment':

neutral_sentiments_per_department[department] = count

    # Print the dictionaries to debug
    print("Positive Sentiments:", positive_sentiments_per_department)
    print("Negative Sentiments:", negative_sentiments_per_department)
    print("Neutral Sentiments:", neutral_sentiments_per_department)

    # Extract data for plotting

positive_sentiments_for_plot = [positive_sentiments_per_department.get(dep, 0) for dep in department_names_for_plot]

negative_sentiments_for_plot = [negative_sentiments_per_department.get(dep, 0) for dep in department_names_for_plot]

neutral_sentiments_for_plot = [neutral_sentiments_per_department.get(dep, 0) for dep in department_names_for_plot]
```

```
    # Check if the request expects plain text (for export)
    export_data = {
        'staff_members': staff_members,
        'feedback_records': feedback_records,
        'selected_staff_id': selected_staff_id,
        'selected_staff_name': selected_staff_name,
        'start_date': start_date,
        'end_date': end_date,
        'selected_department_id': selected_department_id,
        'selected_department_name': selected_department_name,
        'department_names': department_names,
        'average_courtesy': average_courtesy,
        'average_quality': average_quality,
        'average_timeless': average_timeless,
        'average_efficiency': average_efficiency,
        'average_cleanliness': average_cleanliness,
        'average_comfort': average_comfort,
        'total_records': total_records,
        'total_average': total_average,
        'sentiment_counts': sentiment_counts,
        'total_sentiments': total_sentiments,
        'department_names_for_plot': department_names_for_plot, # Added for plotting
        'record_counts_for_plot': record_counts_for_plot, # Added for plotting
        'positive_sentiments': positive_sentiments_for_plot,
        'negative_sentiments': negative_sentiments_for_plot,
        'neutral_sentiments': neutral_sentiments_for_plot,
    }
    return render (request, 'AdminDashboard.html', export_data)
#--->AJAX
def load_staffs(request):
    department_id = request.GET.get('department_id')
    staffs = Staff.objects.filter(department_id=department_id).all()
    return render(request, 'staff_dropdown_list_options.html', {'staffs': staffs})
#--->end of AJAX <dev elgene>
# <dev elgene>
@login_required
def change_username(request):
    if request.method == 'POST':
        current_username = request.POST['current_username']
        new_username = request.POST['new_username']
        user = User.objects.get(username=current_username)
        if user.check_password(request.POST['current_password']):
            user.username = new_username
            user.save()
            return JsonResponse({'success': True})
        else:
            return JsonResponse({'success': False, 'error': 'Current password is incorrect'})
    else:
        return JsonResponse({'success': False, 'error': 'Invalid request method'})
# <dev elgene>
@login_required
def change_password(request):
    if request.method == 'POST':
```

```
        current_username = request.POST.get('current_username')
        current_password = request.POST.get('current_password')
        new_password = request.POST.get('new_password')

        user = User.objects.get(username=current_username)

        if user.check_password(current_password):
            form = PasswordChangeForm(user, {
                'new_password1': new_password,
                'new_password2': new_password,
                'old_password': current_password,
            })
            if form.is_valid():
                form.save()
                update_session_auth_hash(request, user)
                return JsonResponse({'success': True, 'message': 'Password successfully changed'})
            else:
                # Form errors include more specific error messages
                errors = form.errors
                return JsonResponse({'success': False, 'error': errors})
        else:
            return JsonResponse({'success': False, 'error': 'Current password is incorrect'})
    else:
        return JsonResponse({'success': False, 'error': 'Invalid request method'})
# <dev elgene>
@login_required
def DepartmentDashboard(request):
    gene = request.session.get('gene', '')  # Retrieve the data

    # this will return the user to the login page if they try to go to the link of the  of the dashbaord
    # without loging in because there will be no seesion id for department id
    if gene=='':
        return redirect('login')

    # You can access the data here
    departmentID = gene
    department = Department.objects.get(id=departmentID)
    staff_members = Staff.objects.filter(department_id=departmentID)
    feedback_records = Feedback.objects.filter(department=department)
    selected_staff_id = request.POST.get('attendingStaff')
    start_date = request.POST.get('start-date')
    end_date = request.POST.get('end-date')

    if selected_staff_id and selected_staff_id != "":
        staff = Staff.objects.get(id=selected_staff_id)
        feedback_records = feedback_records.filter(staff=staff)

        #get staff name

        selected_staff_name=staff
    else:

        selected_staff_name="Over All Evaluation"

    if start_date and end_date:
        # Parse the start and end dates as datetime objects
        start_date = datetime.strptime(start_date, '%Y-%m-%d')
        end_date = datetime.strptime(end_date, '%Y-%m-%d')

        # Filter feedback records by date range
```

```
        feedback_records = feedback_records.filter(dateOfVisit__gte=start_date, dateOfVisit__lte=end_date)

    # Initialize variables for calculating averages <dev elgene>
    total_records = 0
    total_courtesy = 0
    total_quality = 0
    total_timeless = 0
    total_efficiency = 0
    total_cleanliness = 0
    total_comfort = 0
    sentiments = []

    for feedback in feedback_records:
        total_records += 1
        total_courtesy += (feedback.courtesy) * 20 # Convert to a scale of 0 to 100 <dev elgene>
        total_quality += (feedback.quality) * 20
        total_timeless += (feedback.timeless) * 20
        total_efficiency += (feedback.efficiency) * 20
        total_cleanliness += (feedback.cleanliness) * 20
        total_comfort += (feedback.comfort ) * 20

        sentiments.append(feedback.sentiment)

    # Calculate average values for areas of concern <dev elgene>
    average_courtesy = total_courtesy / total_records if total_records > 0 else 0
    average_quality = total_quality / total_records if total_records > 0 else 0
    average_timeless = total_timeless / total_records if total_records > 0 else 0
    average_efficiency = total_efficiency / total_records if total_records > 0 else 0
    average_cleanliness = total_cleanliness / total_records if total_records > 0 else 0
    average_comfort = total_comfort / total_records if total_records > 0 else 0

    # Calculate the total average <dev elgene>
    total_average = (average_courtesy + average_quality + average_timeless + average_efficiency + average_cleanliness + average_comfort) / 6

    # Count sentiments
    sentiment_counts = dict(Counter(sentiments))
    total_sentiments = sum(sentiment_counts.values())

    selected_staff_id = request.POST.get('attendingStaff')
    start_date = request.POST.get('start-date')
    end_date = request.POST.get('end-date')

    return render(request, 'DepartmentDashboard.html', {
        'staff_members': staff_members,
        'feedback_records': feedback_records,
        'selected_staff_id': selected_staff_id,
        'average_courtesy': average_courtesy,
        'average_quality': average_quality,
        'average_timeless': average_timeless,

        'average_efficiency': average_efficiency,

        'average_cleanliness': average_cleanliness,
        'average_comfort': average_comfort,
        'total_records': total_records,
        'total_average': total_average,
        'sentiment_counts': sentiment_counts,
        'total_sentiments': total_sentiments,

        'selected_staff_name': selected_staff_name,

        'start_date':start_date,

        'end_date':end_date,
```

```
'department':department,
    })
# <dev elgene>
```