

**STUDENT ENGAGEMENT ANALYSIS IN VIRTUAL CLASS USING NATRUAL LANGUAGE PROCESSING**

**A PROJECT REPORT**

***Submitted by***

**DHANUSH G (1920102028)**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***In***

**COMPUTER SCIENCE AND ENGINEERING**

**SONA COLLEGE OF TECHNOLOGY**

**(An Autonomous Institution)**

**SALEM-636005**

**ANNA UNIVERSITY: CHENNAI-600 025 MAY 2024**

A blue logo with a book and a factory

Description automatically generated

**STUDENT ENGAGEMENT ANALYSIS IN VIRTUAL CLASS USING NATRUAL LANGUAGE PROCESSING**

**A PROJECT REPORT**

***Submitted by***

**DHANUSH G (1920102028)**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***In***

**COMPUTER SCIENCE AND ENGINEERING**

**SONA COLLEGE OF TECHNOLOGY**

**(An Autonomous Institution)**

**SALEM-636005**

**ANNA UNIVERSITY: CHENNAI-600 025**

**MAY 2024**

**SONA COLLEGE OF TECHNOLOGY, SALEM-5**

**(AUTONOMOUS)**

AFFILIATED TO ANNA UNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

Certified that this project report **“STUDENT ENGAGEMENT ANALYSIS IN VIRTUAL CLASS USING NATRUAL LANGUAGE PROCESSING”** is the Bonafide work of **“DHANUSH G (1920102028)”** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **DR. B. SATHIYABHAMA**  **PROFESSOR,** | **MR. M BALAMURUGAN**  **ASSISTANT PROFESSOR** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Department of CSE | Department of CSE |
| Sona College of Technology, Salem – 636005. | Sona College of Technology, Salem – 636005. |

Submitted for the project viva-voce examination held on ………...

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**CONFERENCE CERTIFICATE**

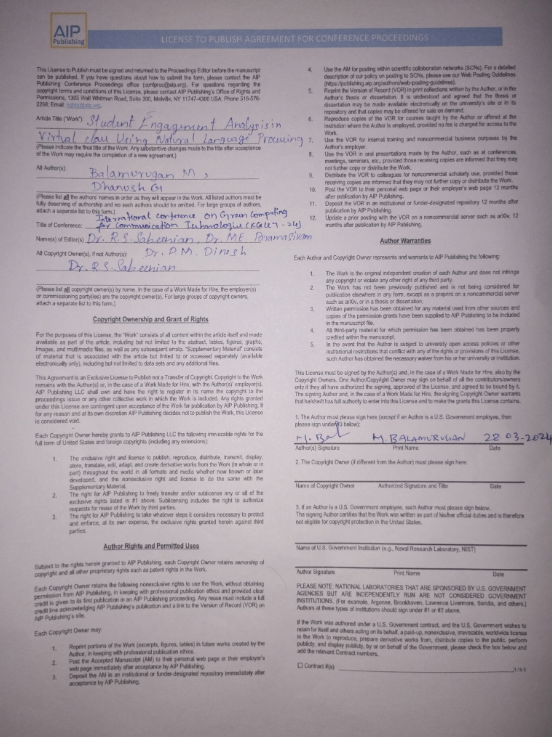
****

**INTERNSHIP COMPLETION CERTIFICATE**

A close-up of a letter

Description automatically generated

**PROCEEDINGS**

****

****

**ACKNOWLEDGMENT**

First and foremost, I would like to express my gratefulness to our honorable Chairman **Sri. C Valliappa** our Vice Chairman **Sri. Chocka Valliappa** and **Sri Thyagu Valliappa** and the management of Sona College of Technology for bring me constant encouragement throughout this course.

My Sincere thanks goes to **Dr. S. R. R. Senthil Kumar**, Principal. Sona College of Technology, who has motivated me in my entire endeavors. I wholeheartedly thank Dr**. B. Sathiyabhama**, Professor and Head Department of Computer Science and Engineering, Sona College of Technology Salem for giving constant encouragement and rendering all kinds of support throughout the course.

I use this opportunity to express my deepest gratitude and special thanks to my project guide **Mr. M. Balamurugan,** Assistant Professor Department of Computer Science and Engineering Sona College of Technology.

Special thanks go to my class counsellor **Ms. K. S. Jayareka,** Assistant Professor, Department of Computer Science and Engineering, Sona College of Technology.

I would like to thank my parents and all my friends from the bottom of my heart who were always present to help me out and make this project a success. Last but not the least, I would like to express my heartiest thanks and gratefulness to almighty God for his divine blessings, which helped me complete the final year project successfully This project was made possible because of inestimable inputs from everyone involved, directly or indirectly.

**ABSTRACT**

The shift to digital education has become imperative, particularly during pandemic situations, leading to an increased reliance on online classes. However, teachers face challenges in gauging student involvement and understanding in virtual environments. This research explores the use of NLP techniques, including the BERT model for understanding relationships among words and customized Jaccard similarity for identifying overlapping words between paragraphs, to address these challenges. We propose a method where teachers make notes about lectures and encourage students to take notes in each class. These notes are then analyzed using NLP algorithms to trace each student's overall understanding and track how their comprehension evolves over time. The study aims to evaluate how students' understanding and involvement in virtual classes can be enhanced through NLP analysis. We present a portfolio that visualizes the overall understanding graph of each student and their understanding score for each class, providing insights for improving virtual teaching methodologies and student learning outcomes. In addition to its primary application in enhancing virtual education, the proposed model serves as a versatile tool with broader utility across diverse domains. Among its manifold capabilities, the model can be harnessed as an effective answer evaluation system, facilitating the objective assessment of responses in various contexts.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ACKNOWLEDGMENT** |  |
|  | **ABSTRACT** |  |
|  | **LIST OF FIGURES** |  |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 PROBLEM STATEMENT | 2 |
|  | 1.3 OBJECTIVE OF PROJECT | 2 |
|  | 1.4 SCOPE OF PROJECT | 3 |
| **2** | **LITERATURE SURVEY** | **5** |
|  | 2.1 RESEARCH WORK | 5 |
| **3** | **SYSTEM ANAYSIS** | **7** |
|  | 3.1 EXISTING SYSTEM | 7 |
|  | 3.1.1 DISADVANTAGES | 7 |
|  | 3.2 PROPOSED SYSTEM | 8 |
|  | 3.2.1 OVERVIEW | 8 |
|  | 3.2.2 MODULES AND THEIR FUNCTIONALITIES | 10 |
|  | 3.2.3 ADVANTAGES | 11 |
|  | 3.3 SYSTEM REQURIMENTS | 13 |
|  | 3.3.1 HARDWARE REQUIRMENTS | 13 |
|  | 3.3.2 SOFTWARE REQURIMENTS | 13 |
| **4** | **DESIGN AND IMPLEMENTATION** | **14** |
|  | 4.1 METHODOLIGIES | 14 |
|  | 4.1.1 DEDICATED SPECIALIZED  SYSTEM FOR NOTES UPLOAD | 14 |
|  | 4.1.2 DOUBT CLARIFICATION  SYSTEM | 14 |
|  | 4.1.3 GUIDANCE PROVIDING  SYSTEM | 14 |
|  | 4.1.4 LSTM MODEL FOR  VALIDATING HUMAN-GENERATED  TEXT | 15 |
|  | 4.1.5 BERT MODEL FOR  EXTRACTING UNDERSTANDING   SCORE | 15 |
|  | 4.1.6 GRAMMAR CORRECTION  MODEL | 15 |
|  | 4.1.7 VISUALIZATION PORTFOLIO | 15 |
|  | 4.2 ARCHITECTURE DIAGRAM | 16 |
|  | 4.3 CLASS DIAGRAM | 17 |
|  | 4.4 USE CASE DIAGRAM | 18 |
|  | 4.5 SEQUECE DIAGRAM | 19 |
|  | 4.6 FEASIBILTY STUDY | 20 |
|  | 4.6.1 TECHNICAL FEASIBILTY | 20 |
|  | 4.6.2 ECONOMIC FEASIBILTY | 21 |
|  | 4.6.3 BEHAVIORAL FEASIBLITY | 21 |
| **5** | **CONCLUSION AND FUTURE WORKS** | **22** |
|  | 5.1 CONCLUSION | 22 |
|  | 5.2 FUTURE WORKS | 23 |
|  | **APPENDIX** | **26** |
|  | 1. SOURCE CODE | 26 |
|  | 1. SCREENSHOTS | 31 |
|  | **REFERENCES** | **40** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 4.1 | ARCHITECTURE DIAGRAM | 16 |
| 4.2 | CLASS DIAGRAM | 17 |
| 4.3 | USE CASE DIAGRAM | 18 |
| 4.4 | SEQUENCE DIAGRAM | 19 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

The way we learn has changed a lot with everything going digital, especially during tough times like the recent pandemic. Online classes have become the new normal, but they come with their own challenges, especially for teachers. One big challenge is figuring out how well students are really understanding and engaging with the lessons in virtual classes. To tackle this challenge, our research dives into the world of Natural Language Processing (NLP), a fancy term for how computers understand human language. We use NLP to analyze the notes students take in online classes, giving us a peek into how well they are understanding and participating in the lessons. One cool tool we use is called the BERT model. It is like a language detective, sniffing out the meanings behind words in the notes. By using BERT, we can see how much of the teacher's content is getting through to each student. This helps teachers adjust their teaching to make sure everyone is on the same page. Another trick up our sleeve is the customized Jaccard similarity. This one helps us spot if students are copying from each other by looking at the words they use in their notes. It is like plagiarism police for online classes, helping keep things fair and honest. To make things even more interesting, we've created a special website for teachers. This website, made with a tool called Streamlit. It stores all the notes and analyses them using an SQLite database. Then, we use cool visual tools like Matplotlib and Plotly to show the data in easy-to-understand graphs and charts. Our goal is to help teachers better understand how their students are doing in online classes. With our research, we hope to make online learning more engaging and effective for everyone involved, even when we cannot be in the same room together.

* 1. **PROBLEM STATEMENT**

In the current landscape, conventional systems for assessing student engagement in virtual classes primarily rely on camera monitoring and sensor-based tracking, while NLP techniques focus on participation analysis. However, these approaches encounter significant limitations, including privacy concerns, high sensor costs, and a lack of direct interaction in online classrooms. Moreover, existing models often fall short in delivering satisfactory performance. To address these challenges, we introduce a groundbreaking platform that revolutionizes the assessment paradigm. By leveraging student-generated notes as a rich data source, our platform offers a unique solution for gauging engagement and comprehension. Unlike traditional methods, our system ensures privacy, eliminates excessive costs, and fosters meaningful interaction. With our portfolio, students can seamlessly upload notes, enabling teachers to gain invaluable insights efficiently through advanced multi-model analysis. This innovative approach promises to redefine the assessment landscape, providing educators with unparalleled tools to enhance virtual teaching effectiveness and student learning outcomes.

* 1. **OBJECTIVE OF PROJECT**

Our project aims to transform how student engagement and comprehension are assessed in virtual classrooms by harnessing the power of Natural Language Processing (NLP) on student-generated notes. Our objectives are multifaceted: first, to create a platform that efficiently processes these notes, providing valuable insights into student understanding and participation. Second, we seek to employ advanced NLP models, such as BERT and customized similarity algorithms, to analyze the textual content of these notes comprehensively. Third, we endeavor to ensure privacy and cost-effectiveness by eliminating the need for intrusive monitoring systems and expensive sensors. Fourth, we aim to foster meaningful interaction in online classrooms by empowering teachers with actionable insights derived from student-generated data. Our ultimate goal is to enhance virtual teaching effectiveness and student learning outcomes through informed pedagogical adjustments based on the analysis of these notes. Additionally, we're committed to delivering a user-friendly interface for both students and teachers, enabling seamless uploading, access, and interpretation of analyzed notes and insights. Finally, we aim to establish a robust portfolio demonstrating the effectiveness of our approach in elevating the overall quality of virtual education experiences.

* 1. **SCOPE OF PROJECT**

The scope of our project encompasses various dimensions aimed at revolutionizing the assessment and enhancement of virtual education experiences through innovative use of Natural Language Processing (NLP) techniques on student-generated notes. Key components of our project scope include:

**Platform Development:** We will develop a user-friendly platform that allows students to upload their notes and enables teachers to access and interpret analyzed insights efficiently.

**NLP Integration:** The project involves integrating advanced NLP models, such as BERT and customized similarity algorithms, to analyze the textual content of student notes comprehensively.

**Privacy and Cost-effectiveness**: We will ensure that the platform upholds privacy standards and offers a cost-effective alternative to traditional monitoring systems and sensors.

**Insightful Analysis:** The platform will provide valuable insights into student understanding and participation in virtual classes, empowering teachers to make informed pedagogical adjustments.

**User Interface Design:** We will focus on creating an intuitive and visually appealing user interface that facilitates seamless interaction for both students and teachers.

**Portfolio Development:** We plan to develop a robust portfolio showcasing the effectiveness of our approach in elevating the overall quality of virtual education experiences.

**Scalability and Adaptability:** The project will be designed to scale and adapt to different educational settings and subject areas, ensuring its broader applicability.

**LSTM Model for Content Validation:** We will implement a Long Short-Term Memory (LSTM) model to distinguish between notes uploaded by students and those generated by AI. This validation mechanism ensures the authenticity of the content, distinguishing between human-generated and AI-generated notes.

**Career Guidance Section for Teachers:** We will dedicate a section within the platform specifically tailored to career guidance, enabling teachers to provide students with valuable insights and advice on career paths and opportunities. This section will facilitate personalized guidance, helping students make informed decisions about their future endeavors.

**Doubt Assistance Section for Students:** A dedicated section will be created for students to seek clarification and assistance with their doubts and questions directly from teachers. This interactive feature fosters engagement and enables students to receive timely support, enhancing their understanding and learning experience in virtual classrooms.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 RESEARCH WORKS**

Kim et al. (2023) explored the characteristic behaviors of elementary students in a low attention state during online learning sessions using Electroencephalography (EEG) technology. By analyzing EEG data, the authors identified specific indicators such as mouth movements and body posture changes associated with decreased attention levels. This study sheds light on the challenges of maintaining student engagement in virtual classrooms and underscores the potential of EEG technology in monitoring attention states in young learners.

In a study by Zhen et al. (2023), the authors focused on predicting the academic performance of students in online live classroom interactions using Natural Language Processing (NLP) and Deep Learning methods. Analyzing interactions among students and with teachers in a large dataset of over 89,000 students, the authors identified patterns linked to better grades. By leveraging computer models, they successfully determined key factors predictive of academic success, offering valuable insights for educators to provide targeted support in online education settings.

Li et al. (2022) proposed an intelligent stable monitoring algorithm of students' online learning behavior based on user portraits. Their smart system captures learning behavior rates using user portraits, incorporating various attributes such as behavior, interests, and hobbies. The system boasts a high accuracy rate in capturing learning behavior, offering educators actionable insights into student engagement and performance in virtual classrooms.

In another approach, Su et al. (2021) introduced a video analytic in-class student concentration monitoring system designed to detect students' low learning states without compromising their privacy. Utilizing cameras and computers, the system identifies signs of distraction, such as smartphone use, while preserving student privacy by ensuring data remains within the server. This innovative system provides educators with real-time feedback on student engagement, enabling timely interventions to maintain classroom focus.

Finally, Blanco and Moldovan (2021) presented the BERT BiLSTM-Attention Similarity Model, a novel approach to enhancing question-answering systems. Leveraging BERT for question understanding and BiLSTM-Attention to highlight crucial elements within questions, the model employs a unique formula to determine similarity between questions. This model holds promise for improving the accuracy and efficiency of question-answering systems, thereby enhancing the online learning experience for students.

In summary, these studies collectively contribute to advancing our understanding of student engagement, performance prediction, and behavior monitoring in online learning environments. By leveraging cutting-edge technologies and methodologies, researchers are paving the way for more effective and personalized educational experiences in virtual classrooms.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

The existing systems for assessing student engagement and participation in virtual classes predominantly rely on camera monitoring, sensor-based tracking, and NLP techniques. These systems aim to gauge student involvement and comprehension levels during online sessions. However, they suffer from various drawbacks, which can be outlined as follows:

**3.1.1 Drawbacks**

**Privacy Concerns:** The utilization of camera monitoring raises significant privacy concerns among students and may infringe upon their rights to privacy. Students may feel uncomfortable with the constant surveillance in their learning environment, leading to potential distrust in the system.

**High Sensor Costs:** Implementing sensor-based tracking systems incurs high costs, which can be prohibitive for educational institutions, especially those with limited budgets. The expenses associated with purchasing and maintaining these sensors may outweigh the benefits they provide.

**Lack of Conversation in Online Classes:** Despite the use of NLP techniques to analyze text-based interactions, the absence of real-time conversation in online classes limits the depth of insights that can be gleaned from student participation. This lack of interaction may hinder the accuracy of the analysis and the ability to effectively gauge student comprehension.

**Limited Performance:** Existing systems may fail to deliver satisfactory performance in accurately assessing student engagement and comprehension levels. The reliance on static measures such as movement tracking and text analysis may overlook nuanced aspects of student participation and understanding.

**3.2 PROPOSED SYSTEM**

**3.2.1 OVERVIEW**

The system architecture of our streamlit portfolio can be designed using a multi-tier architecture, which separates the system into three main tiers: presentation tier, application tier, and data tier.

**Presentation Tier (Streamlit):** At the presentation tier, we employed Streamlit to develop a user-friendly interface for our plagiarism detection system. Streamlit facilitated seamless interaction with the system, allowing users to input text data and view plagiarism detection results in a clear and intuitive manner. The Streamlit application included features such as text input forms, result visualization, and user feedback mechanisms, ensuring a streamlined user experience.

**Application Tier (Streamlit and Python):** The application tier, powered by Streamlit and Python, housed the business logic and processing logic for plagiarism detection. Python scripts integrated with Streamlit were responsible for implementing the plagiarism detection algorithms and handling user interactions. These scripts utilized customized Jaccard similarity measures to compare text data retrieved from the SQLite database, effectively identifying overlapping words between paragraphs to detect plagiarism.

**Data Tier (Python with SQLite):** For the data tier, Python with SQLite served as the foundation for storing and managing text data utilized in plagiarism detection. The SQLite database stored text documents and provided efficient retrieval mechanisms for comparison purposes. Python scripts interfaced with the SQLite database, retrieving text data for comparison and facilitating seamless data management operations.

**Plagiarism Detection:** Within the application tier, Python scripts implemented customized Jaccard similarity measures to detect plagiarism effectively. These scripts interfaced with the SQLite database, retrieving text data for comparison and calculating similarity scores based on overlapping words between paragraphs. The plagiarism detection algorithms facilitated accurate identification of plagiarized content, ensuring the integrity of the text data used in the system.

**Summary:** By leveraging Streamlit for the presentation tier and Python with SQLite for the application and data tiers, we developed a comprehensive plagiarism detection system. The integration of customized Jaccard similarity measures enabled accurate detection of plagiarized content, ensuring the reliability and effectiveness of the system in identifying instances of plagiarism in text documents. Through seamless interaction and efficient data management, our system provided users with a robust solution for plagiarism detection in educational and academic settings.

**3.2.2 MODULES AND THEIR FUNCTIONALITIES**

**Admin Module:** The admin module serves as the backbone of the system, providing administrators with comprehensive control over various aspects of the platform. Administrators possess the authority to manage teacher details, including creation, updating, and deletion of profiles. Additionally, they have the capability to view teacher details based on specific criteria, facilitating streamlined access to relevant information. Furthermore, administrators can create timetables and allocate subjects to teachers, ensuring efficient scheduling of classes. The admin module also encompasses functionality for managing student details, allowing administrators to manage student details, including creation, updating, and deletion of profiles

**Teacher Module:** Teachers access the system through a dedicated login portal tailored to their needs. Within this module, teachers are presented with three distinct sections to facilitate their roles effectively. Firstly, teachers can upload their lecture notes, ensuring seamless dissemination of course materials to students. Additionally, teachers have the option to initiate model training, enabling the system to learn from uploaded notes and provide personalized guidance to students. Moreover, teachers can utilize a dedicated section to provide guidance to selected students, fostering individualized support and mentorship. Furthermore, the module includes a doubt clarification section, allowing teachers to address student queries promptly and comprehensively. Finally, teachers can leverage an insights section to analyze data collected from student notes, enabling informed decision-making and instructional improvement.

**Student Module:** The student module offers a user-friendly interface designed to cater to students' academic needs effectively. Upon logging in, students are presented with three distinct sections tailored to optimize their learning experience. Firstly, students can upload their notes, facilitating seamless collaboration and knowledge sharing. Additionally, students have access to a dedicated guidance section, where they can seek personalized support and mentorship from teachers. Furthermore, students can utilize a dedicated doubt clarification section to seek clarification on concepts and topics covered in class. Importantly, students also have the opportunity to provide feedback on teaching methods or note quality through a fault identification section, ensuring continuous improvement in the learning process. Additionally, special measures are implemented to ensure the anonymity of students when providing feedback, thereby fostering a safe and open learning environment.

**3.2.3 ADVANTAGES**

**Efficient Administration:** The modular architecture streamlines administrative tasks, allowing administrators to manage teacher and student details, create timetables, and allocate subjects effortlessly. This enhances overall operational efficiency and organizational effectiveness.

**Tailored Teacher Experience:** The teacher module offers a comprehensive suite of tools and features tailored to educators' needs. From uploading lecture notes to providing personalized guidance and addressing student doubts, the module empowers teachers to deliver high-quality instruction and support effectively.

**Improved Student Engagement:** The student module provides students with intuitive interfaces and dedicated sections for uploading notes, seeking guidance, and clarifying doubts. This promotes active engagement in the learning process and fosters a conducive environment for academic growth.

**Personalized Learning:** The system's ability to train models based on uploaded notes enables personalized learning experiences for students. By analyzing data and providing insights, the system can offer tailored recommendations and guidance, catering to individual learning needs and preferences.

**Feedback Mechanism:** The fault identification section in the student module allows students to provide feedback on teaching methods or note quality anonymously. This promotes open communication and continuous improvement, fostering a culture of collaboration and feedback-driven learning.

**Data Analysis and Insights:** The system's insights section enables teachers to analyze data collected from student notes, facilitating informed decision-making and instructional improvement. By gaining insights into student engagement and comprehension, educators can refine teaching strategies and optimize learning outcomes.

**3.3 SYSTEM REQUIREMENTS**

**3.3.1 HARDWARE REQUIREMENTS:**

The section of hardware configuration is an important task related to software development insufficient random-access memory may adversely affect the speed and the efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

Hardware configurations:

RAM: 8 GB RAM

Processor: Intel Pentium

Processor Speed:2.0GHZ

Hard Disk:2 TB hard disk

RAM: 8 GB RAM

Processor: Intel Pentium

Processor Speed:2.0GHZ

Hard Disk:2 TB hard disk

RAM: 8 GB RAM

Processor: Intel Pentium

Processor Speed:2.0GHZ

Hard Disk:2 TB hard disk

RAM: 8 GB RAM

Processor: Intel Pentium

Processor Speed:2.0GHZ

Hard Disk:2 TB hard disk

RAM: 8 GB RAM

Processor: Intel Pentium

Processor Speed:2.0GHZ

Hard Disk:2 TB hard disk

RAM: 8 GB RAM

Processor: Intel Pentium

Processor Speed:2.0GHZ

Hard Disk:2 TB hard disk

* Processor: Intel core
* Processor Speed: 2.0GHZ
* Hard Disk: 500GB
* Ram: 16 GB\

**3.3.2 SOFTWARE REQURIMENTS:**

A major element in building a system is the section of compatible software since the software in the market is experiencing geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specifications. The study of requirements specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established.

Software configurations:

Client & Server:

* Look design and Coding: Streamlit[Framework] and Python
* Database: SQLite DB

**CHAPTER 4**

**DESIGN AND IMPLEMENTATION**

**4.1 METHODOLOGIES**

**4.1.1 Dedicated Specialized System for Notes Upload**

This system is designed to the process of uploading lecture notes. It operates based on the class timetable, ensuring that the right notes are uploaded for each subject at the appropriate time.

The system can be integrated with the class schedule, identifying the relevant subject and uploading the corresponding notes.

**Benefits:** Efficient, reduces manual effort, ensures timely availability of materials.

**4.1.2 Doubt Clarification System:**

The doubt clarification system allows students to seek clarification on any topic or concept related to their coursework.

Students can submit their doubts through the platform, and teachers or peers can respond with explanations or solutions.

**Benefits:** Enhances student understanding, fosters interaction, and promotes active learning.

**4.1.3 Guidance Providing System:**

This system assists students by providing guidance on various aspects, such as study techniques, exam preparation, and career choices.

It can offer personalized advice based on individual student profiles and performance.

**Benefits:** Supports student success, encourages self-improvement, and boosts confidence.

**4.1.4 LSTM Model for Validating Human-Generated Text:**

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) used for natural language processing.

The model can validate whether text input (such as student answers or notes) is human-generated or automated.

**Benefits:** Helps prevent plagiarism, ensures authenticity of content.

**4.1.5 BERT Model for Extracting Understanding Score:**

BERT (Bidirectional Encoder Representations from Transformers) is a powerful pre-trained language model.

It can assess the depth of understanding in student-generated text.

**Benefits:** Provides insights into student comprehension, identifies areas for improvement.

**4.1.6 Grammar Correction Model:**

This model automatically corrects grammatical errors in student-submitted text.

It can enhance the quality of notes, assignments, and other written content.

**Benefits:** Improves readability, ensures professional communication.

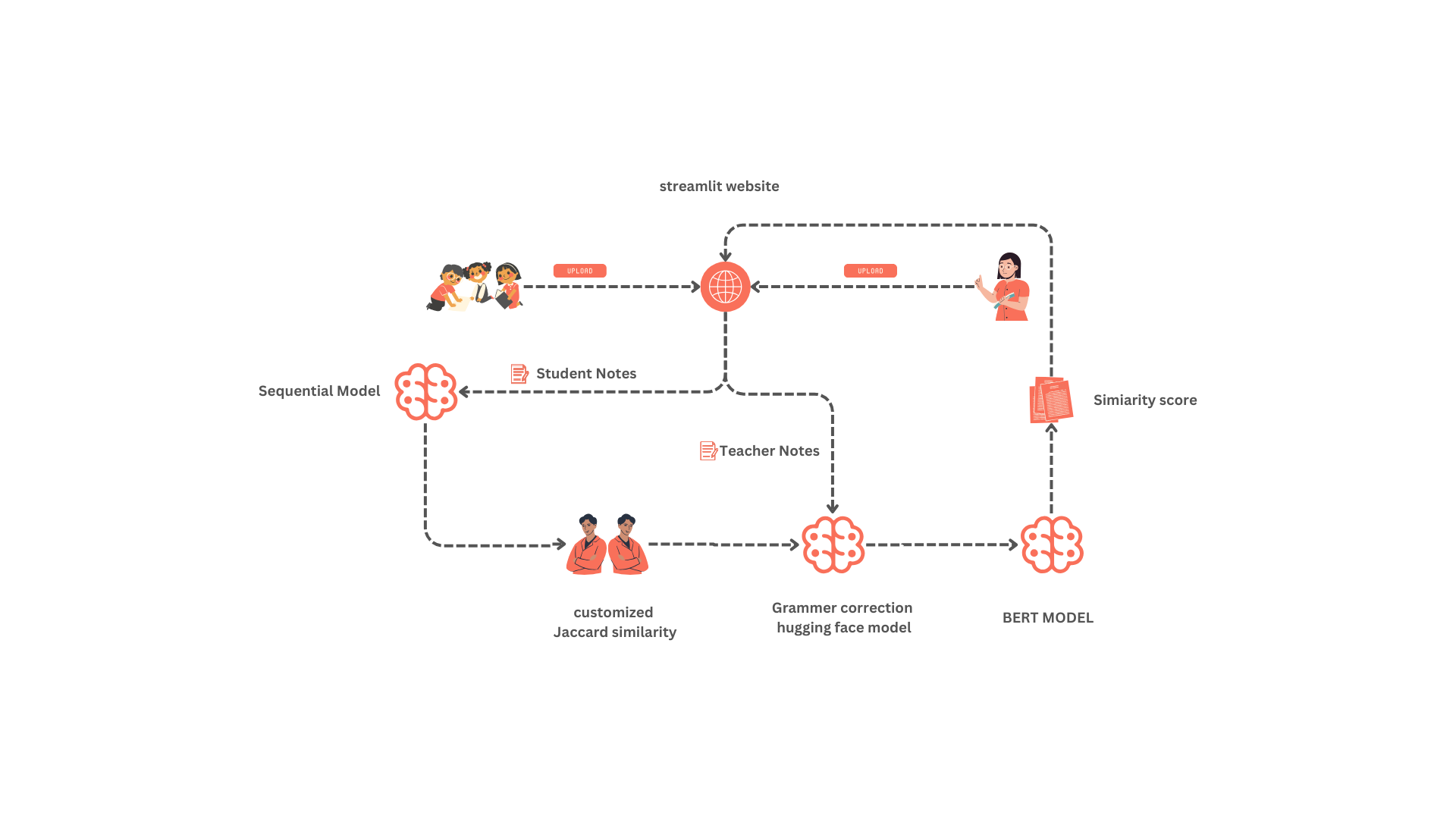
**4.1.7 Visualization Portfolio:**

The visualization portfolio showcases data visualizations related to student performance, attendance, or other relevant metrics.

It can include graphs, charts, and interactive dashboards.

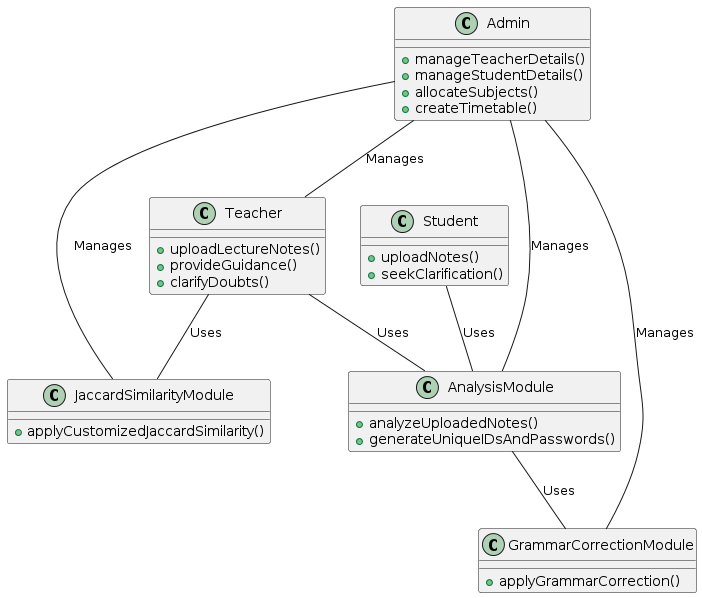
**Benefits:** Makes data more accessible, aids decision-making, and enhances communication.

**4.2 ARCHITECTURE DIAGRAM**



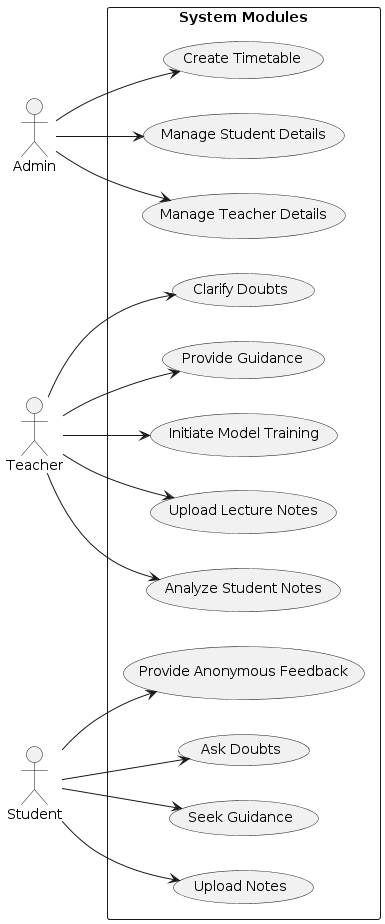
**Fig 4.1 Architecture Diagram**

**4.3 CLASS DIAGRAM**



**Fig 4.2 Class Diagram**

**4.4 USE CASE DIAGRAM**



**Fig 4.3 Use case diagram**

**4.5 SEQUENCE DIAGRAM**

A screenshot of a computer

Description automatically generated

**Fig 4.4 Sequence Diagram**

**4.6 FEASIBILITY STUDY**

A feasibility analysis is a comprehensive evaluation that assesses the operational, financial, and technical aspects of a proposal. It is conducted to determine whether a proposed system can meet user needs using current software and hardware technologies, while also being cost-effective from a business perspective and feasible within budgetary constraints. Feasibility studies should be conducted early in the project lifecycle and are typically relatively inexpensive. The findings of the feasibility study help stakeholders decide whether to proceed with a more detailed analysis or explore alternative solutions. In a feasibility study, the operational aspect examines the need for the proposed system and its alignment with organizational goals. The financial aspect evaluates the cost-effectiveness of the system, including development, implementation, and maintenance costs. The technical aspect assesses the feasibility of implementing the system from a technical standpoint, considering factors such as technology availability and compatibility with existing systems. Overall, a feasibility study is essential for determining the viability of a project, ensuring that it can be developed with available resources, and understanding the cost considerations involved. Facts considered in the feasibility analysis were.

* Technical Feasibility
* Economic Feasibility
* Behavioural Feasibility

**4.6.1 TECHNICAL FEASIBILITY**

The technical feasibility of the project is high, as it leverages well-established technologies and frameworks. The use of Streamlit for the frontend, Python with SQLite for the backend data management, and integration of advanced AI models like BERT, LSTM, and customized similarity measures demonstrate a robust technical foundation. Additionally, the implementation of automated user management systems and model training capabilities showcases the project's technical sophistication. Overall, the project's technical architecture is sound and capable of supporting the desired functionalities effectively.

**4.6.2 ECONOMIC FEASIBILITY**

From an economic perspective, the project demonstrates favorable feasibility. The use of open-source technologies like Streamlit and Python significantly reduces development costs, as these frameworks are freely available and widely supported by the developer community. Additionally, the project's scalability and modular architecture minimize long-term maintenance expenses. Furthermore, the potential benefits of the project, such as improved student engagement and learning outcomes, justify the investment in development and deployment.

**4.6.3 BEHAVIOURAL FEASIBILITY**

The behavioral feasibility of the project is also promising, as it aligns well with the behavior and expectations of stakeholders. Administrators, teachers, and students are accustomed to using digital platforms for communication and collaboration, making the transition to the proposed system relatively seamless. The intuitive user interfaces and personalized features cater to the preferences and needs of each user group, enhancing user satisfaction and adoption. Additionally, the project's emphasis on privacy and security addresses common concerns regarding data protection, further promoting user trust and acceptance.

**CHAPTER 5**

**CONCLUSION AND FUTURE WORKS**

**5.1 CONCLUSION**

In summary, our research has demonstrated the power of combining innovative techniques to enrich text analysis. By blending web scraping with advanced AI models like GPT and BERT, as well as employing meticulous preprocessing, we have significantly boosted our model's capacity to grasp and compare text. Through web scraping, we curated a diverse array of content, ensuring our model learned from a broad spectrum of writing styles and topics. The integration of AI models such as GPT and BERT enabled us to generate a myriad of writing styles for the same content, deepening our model's comprehension of language and context. Preprocessing techniques further refined the text data, guaranteeing our model trained on top-quality, standardized text.Our approach of employing BERT for pattern recognition and BoW for plagiarism detection proved particularly effective in scrutinizing text similarity. BERT's prowess in identifying underlying patterns, coupled with BoW's knack for pinpointing plagiarism by comparing words between paragraphs, provided insightful glimpses into text similarity and integrity. These strategies significantly heightened the reliability and precision of our model's analyses. Our study underscores the transformative potential of integrating these methodologies in text analysis, especially in the realm of online learning. By enhancing our model's ability to discern and assess text similarity, we aspire to pave the way for more sophisticated and efficient text analysis tools across various domains.

**5.2 FUTURE WORKS**

For future developments, our project aims to expand its capabilities to further enhance the virtual learning experience for students. One key area of focus will be the implementation of support for understanding mathematical equations and problems. This enhancement will involve integrating specialized models capable of interpreting and solving mathematical queries posed by students. By leveraging advanced techniques in natural language processing (NLP) and mathematical computation, we aim to provide students with personalized and accurate explanations for a wide range of mathematical concepts and problems. Additionally, we plan to introduce support for other languages to cater to a diverse student population. This expansion will involve integrating multilingual capabilities into our platform, allowing students and teachers to interact and access resources in languages other than English. By providing support for multiple languages, we aim to ensure that our platform is accessible and inclusive for students from diverse linguistic backgrounds. Furthermore, we envision incorporating an LLM (Large Language Model) for doubt clarification, which will offer simplified explanations for complex topics. This advanced model, trained on vast amounts of textual data, will provide students with nuanced and personalized explanations for their queries, enhancing their understanding of challenging concepts. As we continue to develop and refine our platform, we remain committed to improving the virtual learning experience for students by offering personalized support, resources, and guidance. Through ongoing research and development efforts, we aim to address the evolving needs of learners in virtual environments and provide them with the tools and resources they need to succeed academically.

**APPENDICES**

**SAMPLE CODE**

**Models:**

**Bert\_model.py**

from sentence\_transformers import SentenceTransformer  
from sklearn.metrics.pairwise import cosine\_similarity  
from nltk.tokenize import sent\_tokenize  
model = SentenceTransformer('bert-base-nli-mean-tokens')  
def read\_and\_tokenize\_text(text):  
 sentences = sent\_tokenize(text)  
 return sentences  
def calculate\_similarity\_score(text1, text2):  
 sentences\_doc1 = read\_and\_tokenize\_text(text1)  
 sentences\_doc2 = read\_and\_tokenize\_text(text2)  
 embeddings\_doc1 = model.encode(sentences\_doc1)  
 embeddings\_doc2 = model.encode(sentences\_doc2)  
  
 similarity\_scores = cosine\_similarity(embeddings\_doc1, embeddings\_doc2)  
  
 return similarity\_scores.mean()

**grammar\_correction.py**

from happytransformer import HappyTextToText, TTSettings  
from transformers import pipeline  
import spacy  
class Grammer\_Correction:  
 def \_\_init\_\_(self, file):  
 #spacy.cli.download("en\_core\_web\_lg")  
 self.happy\_tt = HappyTextToText("T5", "vennify/t5-base-grammar-correction")  
 self.fix\_spelling = pipeline("text2text-generation",model="oliverguhr/spelling-correction-english-base")  
 self.args = TTSettings(num\_beams=5, min\_length=1)  
 self.nlp = spacy.load("en\_core\_web\_lg")  
 self.text = self.nlp(file.decode("utf-8"))  
  
 def generate\_crted\_text(self, texts):  
 re=[]  
 for text in texts:  
 result = self.happy\_tt.generate\_text(f"grammar:{text}", args=self.args)  
 #print(self.fix\_spelling(result.text,max\_length=4028))  
 re.append(self.fix\_spelling(result.text,max\_length=4028)[0]['generated\_text'])  
 return re  
  
 def get\_sentences(self):  
 sentence = ""  
 sentences = []  
 for i in self.text:  
 if i.is\_sent\_end:  
 if(i.is\_punct):  
 sentence += i.text  
 else:  
 sentence = sentence + " " + i.text  
 sentences.append(sentence)  
 sentence = ""  
 else:  
 sentence = sentence + " " + i.text  
 else:  
 if sentence.strip() != '':  
 sentences.append(sentence)  
 return sentences  
  
 def crt\_the\_grammer(self):  
 corrected\_sentences = self.generate\_crted\_text(self.get\_sentences())  
 return ''.join(corrected\_sentences)  
  
 def make\_grammer\_error\_free(self):  
 return self.crt\_the\_grammer()

**originality\_checker.py**

from tensorflow.keras.models import load\_model  
from tensorflow.keras.preprocessing.text import one\_hot  
from tensorflow.keras.preprocessing.sequence import pad\_sequences  
import numpy as np  
  
class originality\_checker:  
 def \_\_init\_\_(self, text):  
 self.model = load\_model(r'C:\Users\maste\finalyearproject\models\valid\_notes\_check\_model.h5')  
 self.text = text  
 def perform\_pad(self):  
 length = 10129  
 encoded\_vector = self.perform\_onehot()  
 padded\_vector = pad\_sequences([encoded\_vector], maxlen=length, padding='post')  
 return padded\_vector  
 def perform\_onehot(self):  
 vocab\_size = 393537  
 return one\_hot(str(self.text), vocab\_size) # Ensure 'text' is converted to string  
  
 def compute\_model(self):  
 padded\_string = self.perform\_pad()  
 score = self.model.predict(np.array(padded\_string))  
 return score[0][0]

**plagiarism\_check.py**

import spacy  
class plagarism\_check:  
 def \_\_init\_\_(self):  
 self.nlp = spacy.load("en\_core\_web\_lg")  
  
 def tokenize\_and\_clean(self, text):  
 tokens = self.nlp(text.lower())  
 clean\_tokens = []  
 for token in tokens:  
 if not token.is\_stop and not token.is\_punct and not token.is\_space:  
 clean\_tokens.append(token.lemma\_)  
 return clean\_tokens  
  
 def generate\_vectorized\_data(self, text1, text2):  
 word2vec = dict()  
 no = 0  
 for i in text1:  
 if i not in word2vec.keys():  
 word2vec[i] = no  
 no += 1  
 for i in text2:  
 if i not in word2vec.keys():  
 word2vec[i] = no  
 no += 1  
 return word2vec  
  
 def convert\_data\_to\_vectors(self, text1, text2):  
 vector\_representation = self.generate\_vectorized\_data(text1, text2)  
 vdata1 = [vector\_representation[i] for i in text1]  
 vdata2 = [vector\_representation[i] for i in text2]  
 return (vdata1, vdata2)  
  
 def generate\_counter(self, vdata1, vdata2):  
 counter1 = dict()  
 counter2 = dict()  
 for i in vdata1:  
 if i in counter1.keys():  
 counter1[i] += 1  
 else:  
 counter1[i] = 1  
 for i in vdata2:  
 if i in counter2.keys():  
 counter2[i] += 1  
 else:  
 counter2[i] = 1  
 return (counter1, counter2)  
  
 def extract\_similar(self, counter1, counter2):  
 similarity\_extraction = 0  
 for i in counter1:  
 if i in counter2.keys():  
 similarity\_extraction += min(counter1[i], counter2[i])  
 return similarity\_extraction  
  
 def calculate\_common\_score(self, text1, text2):  
 text1\_tokens = self.tokenize\_and\_clean(text1)  
 text2\_tokens = self.tokenize\_and\_clean(text2)  
 counter1, counter2 = self.generate\_counter(text1\_tokens, text2\_tokens)  
 common\_score = self.extract\_similar(counter1, counter2)  
 total\_tokens = max(len(text1\_tokens), len(text2\_tokens))  
 similarity\_score = (common\_score / total\_tokens) \* 100  
 return similarity\_score

**SCREEN SHOT**

**Admin Login**

**A black and grey striped background

Description automatically generated**

**Add Teacher Details**

**A screenshot of a computer

Description automatically generated**

**Edit Teacher Details**

**A screenshot of a computer

Description automatically generated**

**Delete Teacher Details**

**A screenshot of a computer

Description automatically generated**

**View Teacher Details**

**A screenshot of a computer

Description automatically generated**

**Add Student Details**

**A screenshot of a computer

Description automatically generated**

**Edit Student Details**

**A screenshot of a computer

Description automatically generated**

**Delete Student Details**

**A screenshot of a computer

Description automatically generated**

**View Student Details**

**A screenshot of a computer

Description automatically generated**

**Add Timetable Details**

**A screenshot of a computer

Description automatically generated**

**Edit Timetable Details**

**A screenshot of a computer

Description automatically generated**

**View Timetable Details**

**A screenshot of a computer

Description automatically generated**

**Teacher Login**

**A black and grey striped background

Description automatically generated**

**Teacher Profile View Page**

**A screenshot of a computer

Description automatically generated**

**Teacher Notes Upload Page**

**A screenshot of a computer

Description automatically generated**

**Student Login**

**A screenshot of a computer

Description automatically generated**

**Student Profile View Page**

**A screenshot of a computer

Description automatically generated**

**Student Notes Upload Page**

**A screenshot of a computer

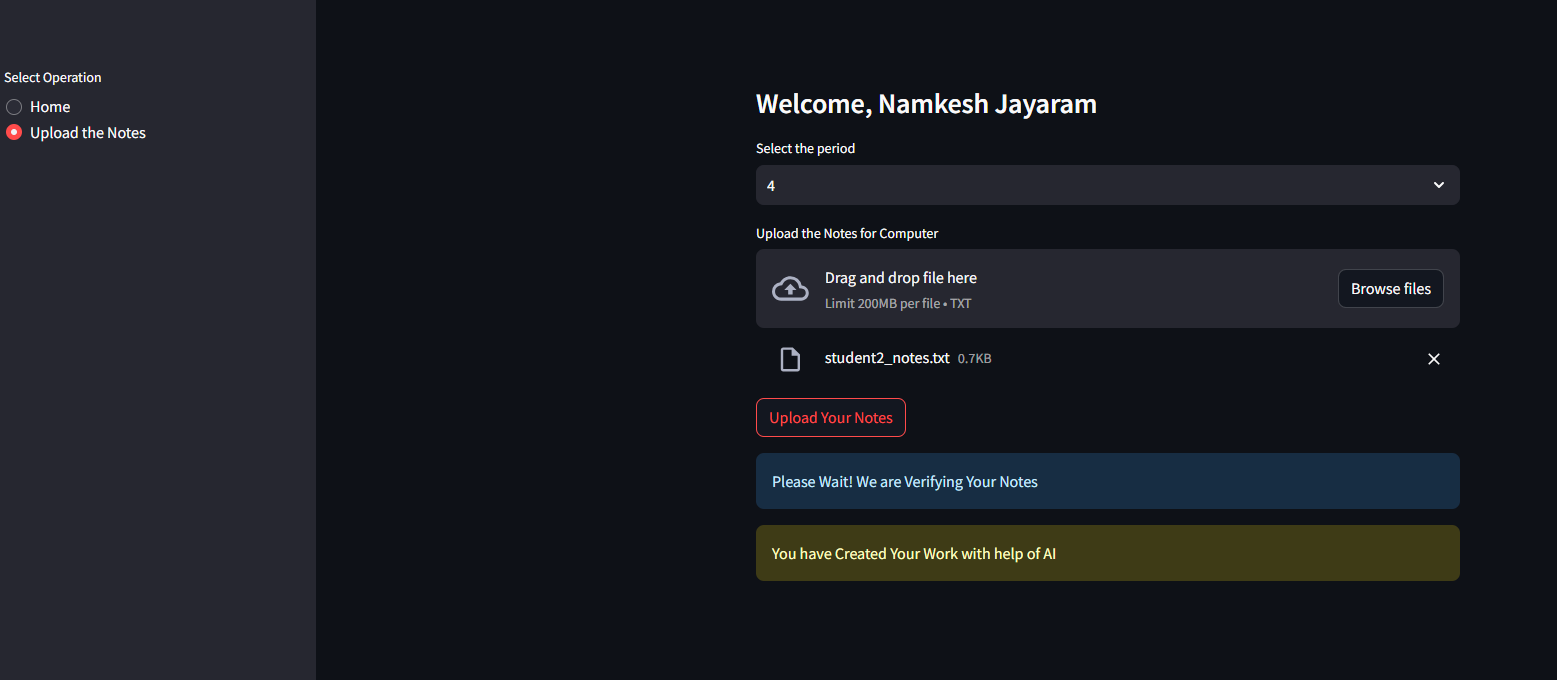
Description automatically generated**

**Malpractice (Type-1)**

**A screenshot of a computer

Description automatically generated**

**Malpractice (Type-2)**

****

**Train the Model Page (Before All Students Upload Notes)**

**A screenshot of a computer

Description automatically generated**

**Train the Model Page (After All Students Upload Notes)**

**A screenshot of a computer

Description automatically generated**

**View Malpractice Details Page**

**A screenshot of a computer

Description automatically generated**

**Title Based Analysis Visualization Page**

**A screenshot of a computer

Description automatically generated**

**Overall, Class Analysis Visualization Page**

**A screenshot of a computer

Description automatically generated**

**Individual Analysis Visualization Page**

**A screenshot of a computer

Description automatically generated**

**Comparative Analysis Visualization Page**

**A screenshot of a computer

Description automatically generated**

**REFERENCES**

1. Suhye Kim, Jung-Hwan Kim, Wooseok Hyung, Suhkyung Shin, Myoung Jin Choi, Dong Hwan Kim, and Chang-Hwan Im. “Characteristic Behaviors of Elementary Students in a Low Attention State During Online Learning Identified Using Electroencephalography.” IEEE Transactions on Learning Technologies, Volume 17, June 2023. DOI: 10.1109/TLT.2023.101682811.
2. Salha M. Alzahrani, Naomie Salim, and Ajith Abraham. “Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods.” IEEE Access, Volume 9, 2021. DOI: 10.1109/ACCESS.2021.30563102.
3. Mu-Chun Su, Chun-Ting Cheng, Ming-Ching Chang, and Yi-Zeng Hsieh. “A Video Analytic In-Class Student Concentration Monitoring System.” IEEE Transactions on Multimedia, Volume 18, Issue 7, July 2016. DOI: 10.1109/TMM.2016.25482983.
4. Eduardo Blanco and Dan Moldovan. “A Semantic Logic-Based Approach to Determine Textual Similarity.” Journal of Artificial Intelligence Research, Volume 24, 2005 4.
5. Krithika and Lakshmi. “Prediction of Academic Performance of Students in Online Live Classroom Interactions — An Analysis Using Natural Language Processing and Deep Learning Methods.” Journal of Social Computing, Volume 4, Issue 1, March 2023. DOI: 10.1007/s42452-023-00195-15.
6. Ahmed Aboutaleb, Ahmed Fayed, Dina Ismail, Nada A. GabAllah, Ahmed Rafea, Nourhan Sakr “BERT BiLSTM-Attention Similarity Model,” IEEE Access, Issue 02 August 2021. DOI: 10.1109/ICAICA52286.2021.9498209
7. Hilal Tekgöz, Halil İbrahim Çelenli, Sevinç İlhan Omurca,” Semantic Similarity Comparison of Word Representation Methods in the Field of Health,” IEEE Access, Issue 13 October 2021, DOI: 10.1109/UBMK52708.2021.9558891
8. Basel Halak, Mohammed El-Hajjar, “Plagiarism detection and prevention techniques in engineering education,” IEEE Xplore, Issue 23 June 2016, DOI: 10.1109/EWME.2016.7496465
9. Wisam A. Qader, Musa M. Ameen, Bilal I. Ahmed “An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges,” IEEE Access, Issue 09 January 2020, DOI: 10.1109/IEC47844.2019.8950616
10. Gang Li, Hai Fu, Yaowen Sun, Yong Zhang “Intelligent Stable Monitoring Algorithm of Students' Online Learning Behavior Based on User Portrait,” IEEE Access, Issue 14 November 2022 DOI: 10.1109/PHM-Yantai55411.2022.9941842