**SMART PROJECT MANAGEMENT SYSTEM**

2021-234

BSc (Hons) in Information Technology

Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 20

**SMART PROJECT MANAGEMENT SYSTEM**

2021-234

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

# DECLARATION

We declare that this is our own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Signature |
| Amarasekara T.N.E. | IT18009446 |  |
| Gamage O.M. | IT18140330 |  |
| Navanjana E.H.D.T.D. | IT18003642 | **A picture containing insect  Description automatically generated** |
| Isurindi H.G.P. | IT18129236 |  |

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

Signature of the supervisor: Date:

# ABSTRACT

Worldwide universities conduct various types of Information Technology Projects. Most of the universities uses common project management solutions which includes industrial quality functionalities. However, there are requirements that doesn’t come with these common project management systems. Automatic Group Generation is one of the major requirement in universities. Project practical teaching has an important significance to enhance students’ practical ability and is also an important part in teaching reform in colleges and universities, however, there is lacking of scientific strategies to grouping students in project practical teaching [5]. Universities doesn’t use any logical way for grouping process. Usually groups will randomly generate or according to a registration list. However, the efficiency and effectiveness of a such group is unpredictable. Therefore, a Computer Based Group Generation system becomes a major requirement for universities.

The Smart Project Management System is proposed to overcome above mentioned requirements as well as most of the common features. System will generate project groups using student Skills, GPA and previous project solutions. With the GitHub integration to the system, grouping process can reach maximum accuracy. However, to calculate the accuracy system need to access various types of skills of the students. Therefore, System will provide facility to manage student profiles by their own, and students can define their very own skills on Programming knowledge. Using this strategy system can use the data that required for the grouping process. Once the process is finished assigned lecturers can check the groups and publish groups among students.

The end result of the solution is a project management system with highly customizable web application. Since the system will an open source solution, Universities can expand the capabilities of the system by their own. Smart project management can be used to improve efficiency and effectiveness in group projects.

*Key words: Project management system, Student grouping, Project contribution manager, Speech Recognition, Peer Review*

# ACKNOWLEDGEMENT

First, we would like to present our deep gratitude for the endless support and guidance helped by our supervisor Ms. Uthpala Samarakoon (Senior Lecturer - Faculty of Computing SLIIT) for her valuable and constructive suggestions during the planning and development of this research work. Her guidance and support helped us the way to get into competitions and show to our talents and complete this SMART Project Management System. Also we would like to give second gratitude to our co-supervisor Ms. Kugathasan Archchana (Lecturer - Faculty of Computing SLIIT) for her encouragement and much needed motivation. Without our supervisors’ major contribution, we would have never obtained this accomplishment.

Not forgetting all the contributors who participated by answering our questionnaires, filling google forms and testing the solution. Furthermore, we would like to thank our parents for their patience, time and providing resources to acquire the needful and for other expenses. Special thanks to our team mates who worked as a team to make this research more effective and success. Thank you for your time, encouragement, and recommendations.

**TABLE OF CONTENTS**

[DECLARATION i](#_Toc85054729)

[ABSTRACT ii](#_Toc85054730)

[ACKNOWLEDGEMENT iii](#_Toc85054731)

[LIST OF FIGURES vi](#_Toc85054732)

[LIST OF ABBREVIATION ix](#_Toc85054733)

[LIST OF APPENDICES xi](#_Toc85054734)

[1. INTRODUCTION 1](#_Toc85054735)

[1.1 Background & Literature survey 3](#_Toc85054736)

[1.2 Research Gap 6](#_Toc85054737)

[1.3 Research Problem 7](#_Toc85054738)

[1.4. OBJECTIVES 9](#_Toc85054739)

[1.4.1 Main Objectives 9](#_Toc85054740)

[1.4.2 Specific Objectives 9](#_Toc85054741)

[2. METHODOLOGY 11](#_Toc85054742)

[2.1 Feasibility Study 12](#_Toc85054743)

[2.2 Requirement Gathering and Analysis 13](#_Toc85054744)

[2.3 Design 15](#_Toc85054745)

[2.4 Implementation 16](#_Toc85054746)

[2.4.1 Group Generation 16](#_Toc85054747)

[2.4.2 GitHub Contribution Predictor 19](#_Toc85054748)

[2.4.3 Client Meeting tracker 24](#_Toc85054749)

[2.4.4 Peer Review 28](#_Toc85054750)

[2.5 Software Solution Development Components 47](#_Toc85054751)

[2.5.1 Web Application Development 47](#_Toc85054752)

[2.5.2 API development 47](#_Toc85054753)

[2.5.3 Database Handling 48](#_Toc85054754)

[2.5.4 Testing 49](#_Toc85054755)

[2.5.5 Deployment 61](#_Toc85054756)

[3. RESULTS AND DISCUSSION 62](#_Toc85054757)

[3.1 Results 62](#_Toc85054758)

[3.1.1 Group Generation 62](#_Toc85054759)

[3.1.2 GitHub prediction results 63](#_Toc85054760)

[3.1.3 Client Meeting tracker 64](#_Toc85054761)

[3.1.4 Peer Review 66](#_Toc85054762)

[3.2 Research Findings 74](#_Toc85054763)

[3.2.1 Group Generation 74](#_Toc85054764)

[3.2.2 GitHub prediction and progress tracking 75](#_Toc85054765)

[3.2.3 Peer Review 76](#_Toc85054766)

[3.2.4 Client Meeting tracker 77](#_Toc85054767)

[3.3 User Interfaces 78](#_Toc85054768)

[3.4 Discussion 83](#_Toc85054769)

[4. SUMMARY OF THIS RESEACH CONTRIBUTION 84](#_Toc85054770)

[5. CONCLUSION 86](#_Toc85054771)

[6. REFERENCES 87](#_Toc85054772)

[7. APPENDICES 91](#_Toc85054773)

[Appendix – A: Sample questionnaire 91](#_Toc85054774)

[Appendix – B: Sample questionnaire response 92](#_Toc85054775)

[Appendix - C: Plagiarism Score 01 93](#_Toc85054776)

[Appendix - D: Plagiarism Score 02 94](#_Toc85054777)

[Appendix - E: Plagiarism Score 03 95](#_Toc85054778)

[Appendix - F: Plagiarism Score 04 96](#_Toc85054779)

[Appendix - G: Plagiarism Score 05 97](#_Toc85054780)

[Appendix - H: Plagiarism Score 06 98](#_Toc85054781)

# LIST OF FIGURES

[Figure 1: Feature Comparison with the existing solutions and proposed system 6](#_Toc85052392)

[Figure 2: SDLC Life Cycle 11](#_Toc85052393)

[Figure 3: Breakdown Structure 12](#_Toc85052394)

[Figure 4: System Architecture Diagram 15](#_Toc85052395)

[Figure 5: Data cleaning process code segment 17](#_Toc85052396)

[Figure 6: Clustering process code segment 17](#_Toc85052397)

[Figure 7: Decision Tree code segment 18](#_Toc85052398)

[Figure 8: Prediction for a single user 21](#_Toc85052399)

[Figure 9: Describe of data and a chart with previous vs this month 22](#_Toc85052400)

[Figure 10: sample data set 23](#_Toc85052401)

[Figure 11: Data cleaning 23](#_Toc85052402)

[Figure 12: Select a model and train it 24](#_Toc85052403)

[Figure 13: Fine Tuning the Model 24](#_Toc85052404)

[Figure 14: meeting platform 25](#_Toc85052405)

[Figure 15: meeting platform after Answering the call 26](#_Toc85052406)

[Figure 16: shows the live voice to text conversion 27](#_Toc85052407)

[Figure 17: show the recorded video of the meeting 27](#_Toc85052408)

[Figure 18: downloaded report 28](#_Toc85052409)

[Figure 19: Student review path structure 29](#_Toc85052410)

[Figure 20: Review types and usage 29](#_Toc85052411)

[Figure 21: Numeric review scale description 30](#_Toc85052412)

[Figure 22: Seven steps of machine learning approach 31](#_Toc85052413)

[Figure 23: Two major paths of the main algorithm 31](#_Toc85052414)

[Figure 24: Access range comparison in the cloud space 33](#_Toc85052415)

[Figure 25: Access range comparison in the cloud space 34](#_Toc85052416)

[Figure 26: Sentiment analysis data preparation 35](#_Toc85052417)

[Figure 27: The text data preprocessing framework 36](#_Toc85052418)

[Figure 28: Importance of noise removal process 37](#_Toc85052419)

[Figure 29: Standard Normalization 38](#_Toc85052420)

[Figure 30: Duplicate processes to validations 39](#_Toc85052421)

[Figure 31: Stemming -suffix removal 40](#_Toc85052422)

[Figure 32: Stemming -prefix removal 40](#_Toc85052423)

[Figure 33: Stemming -infix removal 40](#_Toc85052424)

[Figure 34: Stemming -circumfix removal 41](#_Toc85052425)

[Figure 35: Lemmatization process 42](#_Toc85052426)

[Figure 36: LDA topic model process 43](#_Toc85052427)

[Figure 37: LDA Multicore process 44](#_Toc85052428)

[Figure 38: Perplexity score 45](#_Toc85052429)

[Figure 39: Umass calculation 46](#_Toc85052430)

[Figure 40: Export the system 48](#_Toc85052431)

[Figure 41: Database scripts 49](#_Toc85052432)

[Figure 42: Sensitivity test results part 1 59](#_Toc85052433)

[Figure 43: Sensitivity test results part 2 59](#_Toc85052434)

[Figure 44: Sensitivity test results part 3 60](#_Toc85052435)

[Figure 45: Sensitivity test results part 4 60](#_Toc85052436)

[Figure 46: Sensitivity test results part 5 60](#_Toc85052437)

[Figure 47: Sensitivity test results part 6 60](#_Toc85052438)

[Figure 48: Execute the system 61](#_Toc85052439)

[Figure 49: First group generation Output 62](#_Toc85052440)

[Figure 50: Second group generation output 63](#_Toc85052441)

[Figure 51: final result 64](#_Toc85052442)

[Figure 52: recorded video of the meeting 65](#_Toc85052443)

[Figure 53: Downloaded transcript 65](#_Toc85052444)

[Figure 54: Cv score calculation of the LDA base model 68](#_Toc85052445)

[Figure 55: Umass calculation of the base model 68](#_Toc85052446)

[Figure 56: Initial intertopic distance map 69](#_Toc85052447)

[Figure 57: Term-topic distribution 70](#_Toc85052448)

[Figure 58: Intertopic distance map first iteration 71](#_Toc85052449)

[Figure 59: Term-topic distribution first iteration 72](#_Toc85052450)

[Figure 60: gensim LDA mallet topic generation pattern 73](#_Toc85052451)

[Figure 61: Topic number coverage 76](#_Toc85052452)

[Figure 62: Sentiment score coverage 76](#_Toc85052453)

[Figure 63: Student Survey 78](#_Toc85052454)

[Figure 64: Student Group Generation UI 79](#_Toc85052455)

[Figure 65: create new project environment 79](#_Toc85052456)

[Figure 66: screen for how to create Personal Access Token 80](#_Toc85052457)

[Figure 67: all projects 80](#_Toc85052458)

[Figure 68: single project view 80](#_Toc85052459)

[Figure 69: task dashboard 81](#_Toc85052460)

[Figure 70: prediction tab for each member 81](#_Toc85052461)

[Figure 71: all collaborators 81](#_Toc85052462)

[Figure 72: all commits 82](#_Toc85052463)

**LIST OF TABLES**

[Table 1: CSV template of the reviews 32](#_Toc85052464)

[Table 2: Test case 01 49](#_Toc85052465)

[Table 3: Test Case 02 50](#_Toc85052466)

[Table 4: Test case 03 50](#_Toc85052467)

[Table 5: Test case 04 51](#_Toc85052468)

[Table 6: Test case 05 51](#_Toc85052469)

[Table 7: Test Case 06 52](#_Toc85052470)

[Table 8: Test Case 07 52](#_Toc85052471)

[Table 9: Test Case 08 53](#_Toc85052472)

[Table 10: Test Case 09 53](#_Toc85052473)

[Table 11: Test Case 10 53](#_Toc85052474)

[Table 12: Test case 11 54](#_Toc85052475)

[Table 13: Test case 12 54](#_Toc85052476)

[Table 14: Test case 13 55](#_Toc85052477)

[Table 15: Test case 14 56](#_Toc85052478)

[Table 16: Test case 15 56](#_Toc85052479)

[Table 17: Test case 16 57](#_Toc85052480)

[Table 18: Test case 17 57](#_Toc85052481)

[Table 19: Test case 18 58](#_Toc85052482)

[Table 20: Test case 19 58](#_Toc85052483)

[Table 21: Prediction Overview for User X 64](#_Toc85052484)

[Table 22: LDA base model parameters 66](#_Toc85052485)

[Table 23: Comparison between previous systems 75](#_Toc85052486)

[Table 24: Basic review flow 76](#_Toc85052487)

[Table 25: Description of personnel and facilities. 84](#_Toc85052488)

# LIST OF ABBREVIATION

Abbreviation Description

GPA Grade Point Average

SLIIT Sri Lanka Institute of Information Technology

API Application Programming Interface

SDLC Software Development Life Cycle

XML Extensible Markup Language

ITP Information Technology Project

JS JavaScript

UI User Interface

UX User Experience

JDBC Java Database Connectivity

AWS Amazon Web Services

CSV Comma Separated Values

VCS Version Control System

VC Version Control

PMS Project Management System

REST API RESTful Application Programming Interface

PAT Personal access token

GRA GitHub REST API

URL Uniform Resource Locator

PMI Project Management Institute

PMBOK Project Management Body of Knowledge

NLTK Natural Language Tool Kit

OBE Outcome-Based Education

PBL Problem-Based Learning

HMM Hidden Markov Model

LDA Latent Dirichlet Allocation

JSON JavaScript Object Notation

HTML Hyper Text Markup Language

KPI Key Performance Indicators

OBL Outcome-Based Learning

OBT Outcome-Based Teaching

RE Regular Expressions

NPMI Normalized pointwise mutual Information

PP Perplexity probability

# LIST OF APPENDICES

Appendix Description Page

[Appendix - A Sample questionnaire 91](#_Appendix_–_A:_1)

[Appendix - B Sample questionnaire response 92](#_Appendix_–_B:_1)

[Appendix - C Plagiarism Score 01 93](#_Appendix_-_C:)

[Appendix - D Plagiarism Score 02 94](#_Appendix_-_D:)

[Appendix - E Plagiarism Score 03 95](#_Appendix_-_E:)

[Appendix - F Plagiarism Score 04 96](#_Appendix_-_F:)

[Appendix - G Plagiarism Score 05 97](#_Appendix_-_G:)

[Appendix - H Plagiarism Score 06 98](#_Appendix_-_H:)

# 1. INTRODUCTION

According to Project Management Institute [18], project management is the application of knowledge, skills, tools, and techniques to a broad range of activities to meet the requirements of a particular project. Project Management Body of Knowledge (PMBOK Guide) [18] divides the project management process into five essential steps such as project initiation, project planning, project execution, project performance/monitoring, and project closure. Proposed system contains specific tools to manage these phases with suitable modifications for undergraduate project management.

There are several strategies to generate student groups and Automatic Group generation is one of the required feature for universities. However, there are existing systems used to automate group generation, but most of these solutions are standalone applications and not included in any existing project management solutions. If the grouping process is depending on the student’s choices, some groups might have higher advantage and some of the groups have disadvantages such as student groups with lower student count, students with lower grades and skills are in a one group and higher grade students are in another group. This process will create student groups with Homogeneous groups [1]. Primary goal is to reduce those conflicts by grouping student with different skill structure to increase the diverse among the group members and the algorithm will help to avoid over powering and under powering student groups. Existing project management systems doesn’t come with this feature. The proposed system uses specific skill data from the students and using machine learning algorithms and custom developed algorithm for the grouping process.

On the other hand, students usually do not use any version control, and it is a bad practice to avoid version controlling since the project is a group project. It is imperative to maintain secure source codes and manage individual contributions. The problem is students are not familiar with version control. The use of version control systems can solve another problem of underestimating project duration, which can cause incomplete projects within the specified time. So, incorporating version controlling facilities in project management is another essential aspect of project management. Proposed system uses GitHub REST API [19] to collect project data to track the progress and predict future contributions to a project. API [20] can follow the code, commits, User contribution for the relevant project by passing a GitHub Access Token [21].

Meeting Platform is another central part of this research. Students can have meeting with their client. This is the platform to make a connection between the student and the client. When students are doing their meeting, the Speech Recognition algorithm will follow the meeting platform. Students can track client speech as a real-time component. So the student does not need to take notes.

Emotion recognition has attracted increasingly intense interest from researchers from diverse fields. Human emotions can be recognized from facial expressions, speech, behavior (gesture/posture) or physiological signals [22]. The secondary goal of this meeting platform is to combine with the Emotional Recognition component, that the student can identify the client satisfaction as happy with their work or not. When the meeting is over, the student can download a report containing speech recognition texts format and client satisfaction rate for a general client meeting.

Universities use different kinds of approaches for managing their undergraduate projects which will be expected to help them to improve their collaborative learning skills associated with cooperative learning. Cooperative learning is a specific kind of collaborative learning. In cooperative learning, undergraduates work together in small groups on structured assignments. They are individually accountable for their work, and the work of the group as a whole is also assessed. Cooperative groups work face-to-face and learn to work as a team.

But there are some major issues with undergraduate project module structures that affect these collaborative learning environments which universities trying to maintain. one of those issues is the free-riding effect among undergraduates. Most of the time the free-riding effect is based on unbalanced work breakdown among undergraduate’s project timelines. Without monitoring unbalanced work breakdown, they are becoming passive observers rather than active participants in the project's early stages.

In this smart project management solution, we are addressing those issues. We are introducing a peer review feature to reduce and discourage free-riding in the project groups. This review process will be shared between supervisor and the team as a group objective. this will consider as a milestone like other submissions already existing in the current project module. Peer-reviewing their team members to help supervisors to detect weak points of the undergraduates' collaborative skillsets and help to improve those which is the ultimate result of this research study.

## **1.1 Background & Literature survey**

Today, Technology is being used for everything. From day to day life work to advanced researches we use technology. Technology makes everything easy and manageable. Education is one of the major field that need technological improvements to extend the capabilities. In this section, we are discussing how previous researches tried to improve these aspects and how we’re going to improve those experiments.

According to the existing grouping strategies in project practical teaching, teachers often grouping students according to their willingness, or by their enrollment numbers, or dividing them into groups at random [1]. Dividing students based on above strategies is not very accurate and effective. Students have different skills, different knowledge on programming and also different theoretical and practical skills. Using non logical strategies for grouping may badly effect on group management and also may leads to unsuccessful project.

GroupEng is a solution created by researchers to effectively assign student groups by applying multiple User-prioritized Academic and Demographic Factors [2]. GroupEng is a standalone system which provides the capability to group students by their attributes.

In the research conducted by Zhu Wen-qiang and Wan Ben-ting a different approach was used with fuzzy clustering algorithm. They were able to increase the effectiveness of the grouping process by sorting students using selected factors such as field of studies and previous professional curriculums. Each field has given a value and using that value researchers were able to do a weighted calculation and generate groups according to the importance of each field and the role [5]. They were able to increase the effectiveness of the grouping process using fuzzy clustering algorithm and weighted calculations.

There are several version control systems. But GitHub is the most popular version control system in the project management part. Most of the students in universities are using GitHub. In an existing research to predict the Popularity of GitHub Repositories, GitHub users can show appreciation to projects by adding stars to them. Therefore, the number of stars of a repository is a direct measure of its popularity. They have used multiple linear regressions to predict the number of stars of GitHub repositories. These predictions are useful to both repository owners and clients. [23].

Another research on Using Dynamic and Contextual Features to predict Issue Lifetime in GitHub Projects, the lifetime of an issue depends not only on characteristics of the issue itself, but also on the state of the project as a whole. Hence, issue lifetime prediction may benefit from taking into account features capturing the issue's context [24].

According to the facts mentioned above, we realize that we need a system component to Predict Future GitHub Contribution, which can be used to improve Project Quality. Therefore, predict future contributions with percentages and possibilities using GitHub contribution is one of the essential parts for universities. Because using a version control system is mandatory to doing current group projects.

The BBN continuous speech recognition system is similar solution for speech recognition system. They describe BYBLOS, the BBN continuous speech recognition system. The system, designed for large vocabulary applications, integrates acoustic, phonetic, lexical, and linguistic knowledge sources to achieve high recognition performance [25].

Another research presented, the rapid advances in machine learning (ML) and information fusion has made it possible to endow machines/computers with the ability of emotion understanding, recognition, and analysis [22]. Finally, they compare different ML and deep learning algorithms for emotion recognition and suggest several open problems and future research directions in this exciting and fast-growing area of AI.

There is another Emotional Recognition detection system called “A facial expression emotion recognition based human-robot interaction system”. According to that research, a facial expression emotion recognition based human-robot interaction (FEER-HRI) system is proposed, for which a four-layer system framework is designed. The FEER-HRI system enables the robots not only to recognize human emotions, but also to generate facial expression for adapting to human emotions [26].

When it comes to assessing the reviews or feedbacks on various areas, sentiment analysis and opinion mining are major research categories in natural language processing and machine leaning algorithms. Most of those studies mainly focus on classification of the topics and their sentiments, which is trying to define the sentimental polarity of a text which is negative or positive. Sentiment classification methodologies have been used in various datasets in recent years of research including product reviews [28, 27], tweets [25, 29], news articles [30]. The approaches used to categorize sentiments can be divided to methods depend on lexicons [28] or methods depend on machine learning [27, 29], the neural network models [25, 29], have proven significant performance. Sentiment classification techniques, with all those researches we can determine our research area have not considered for the approach of peer-reviews assessing for undergraduate project management processes. Few research studies have tried to automate the prediction of peer reviews' effectiveness and enhance the quality of peer review assessments. Peer review evaluations in these research areas go beyond simply writing a basic research paper .in this research study we are trying to bring those complex review analyses to enhance the active learning environments in universities to improve the efficiency and productivity of student’s work.

According to the above-mentioned facts, it is clear that member grouping plays an important role in project management. Having a right combination of students with variety of skills and knowledge clearly shown a positive effect on project’s success. Also, this research proposed a plan to predict each student's future contributions for a group project with GitHub commits using the Supervised Learning method. Furthermore, in this research, we provide facilities to both speech recognition and Emotional Recognition as a single component. by combining emotional recognition component with the meeting platform both the client and the student can get an idea about the meeting. Also, real-time speech recognition and Emotional recognition will provide a best solution for the client portal. Moreover, this research will be able to bring complex review analyses to enhance the active learning environments in universities to improve the efficiency and productivity of student’s work.

## **Research Gap**

Above mentioned research solutions are targeted for a specific requirement that only satisfy specific dataset. The problem is, is it configurable for our own requirements and is it possible to use these solutions for our own system. Since all above mentioned systems are made for a single task it is bit complex to integrate it to the proposed project management system. The target is to use existing solutions and reconfigure them to satisfy our own requirements. Some of the solutions are complex to be configured and some of them are not enough sufficient to satisfy our requirement. So to implement an efficient and accurate system, we need to study these existing systems to configure and extend capabilities of the existing solutions.

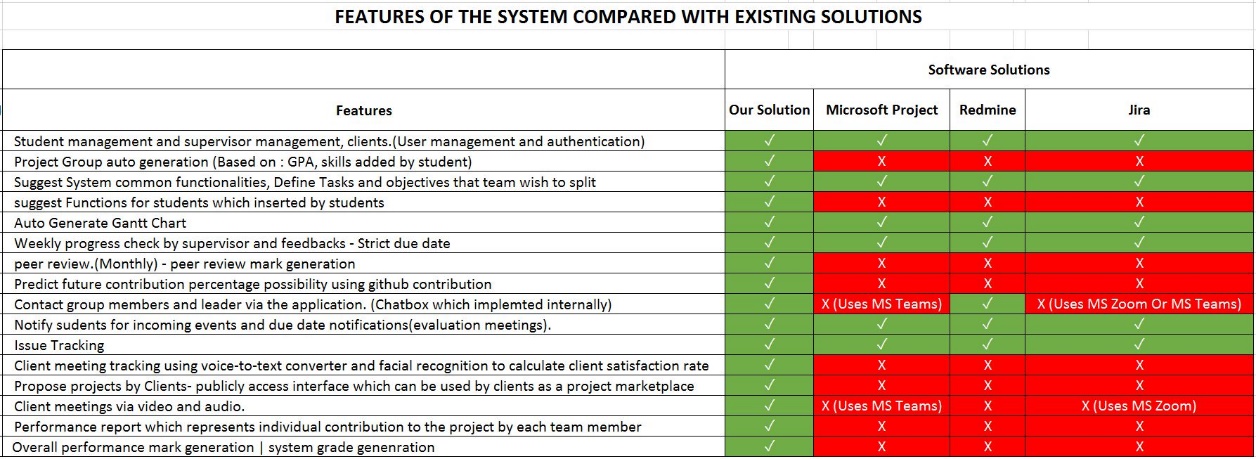


Figure 1: Feature Comparison with the existing solutions and proposed system

Figure 1 shows a table that we created by finding functionalities in the existing project management systems. Cells with ✕ symbol show the functionality is not implemented in the system and cells with ✓ symbol shows the functionality is implemented in the system. As you can see the grouping mechanism is not implemented in neither software solutions and it is a major requirement for any university to improve effectiveness and efficiency of university projects.

## **Research Problem**

Undergraduate project management is challenging when it comes to group projects. Even though there are many existing project management systems such as Microsoft project, Jira, and Redmine, most of them were developed for general purpose. Hence, some important specific features which are useful when managing student projects such as automatic group formation, project tracking and notification generation on project progress are not available in those systems.

One of the major problems in group projects is fair group formation. Self-group formation by students in group project is not successful due to various issues. In such situations a group may have included only the best students in the batch and there is a possibility of left out weak students. Then the weak students may have to form a group which is not successful. In the other hand, when the university is taking the responsibility for grouping, usually the process is random grouping using a student list. There are some online tools which can be used for grouping. Those tools also use random group generation using an existing student list, which is not effective and fair [3]. There are some separate tools to do the process which is not connected with project management tools [4]. Those tools are available in standalone and web-based applications. Hence, a university should use a combination of different tools to accomplish the tasks of project management. Hence there is a requirement of a system where project grouping can be done fairly and effectively.

Another major issue was tracking the project progress. There is no project tracking system and it leads to project failures. In the other hand, students usually do not use any version controlling and it is a bad practice to avoid version controlling since the project is a group project. Version control systems (VCSs) are used to store and reconstruct past versions of program source code [3]. In industry almost all the IT companies use version controlling systems to update their systems and also to carry out their projects and source codes. It is very important to maintain secure source codes and manage individual contributions. Problem is students are not familiar with version controlling. Use of version controlling systems can solve another problem of underestimating project duration which can cause incomplete projects within the specified time. So, incorporating version controlling facility in project management is another important aspect in project management.

Moreover, in existing project management tools there is no specific way to detect the free riders in the group. If the required activities submitted to the system in the given time of the project timeline system, consider it as a successful submission. There are no ways implement to track their performance and progress status with their project team. If the system can detect the free riders in a team, it will lead to reduce group conflicts and effectively differentiate student performance. So, it will be a fair ground to all the project members including project group leader [17].

When the project is for a specific client, it is important to trach the progress of the project by client and give feedback. Also, even if there is a client connected to the project, project coordinators cannot track their meeting details and what they communicated with the client. Students also get into trouble in this situation. Students doesn’t usually take notes in a meeting and it is not necessary if student able to remember all the details and plans that made with client. But they are not. After few days they don’t even know what they are doing. It is better to have any solution to auto generate a report of the meeting and the content. With current systems we have not found any solutions to this matter. Client also find difficulty to track project progress and clients may also blame for the project coordinators for any failures.

## **1.4. OBJECTIVES**

### **1.4.1 Main Objectives**

The main objective of the research is to implement a system that specifically build for management of university projects. Since all other existing project management solutions are developed for common usage, it doesn’t offer some of the required solutions from universities. Such as, Student group generation, GitHub contribution predictor, Client meeting documentation and the peer review functionality. Each component plays a major role in project management. Group generation is responsible for dividing students into groups in order to participate in a group project. GitHub contribution predictor helps to identify incoming deadlines and the possibility of finishing the project in time. Client meeting documentation helps students to automatically document the meeting vocal contents to text content. Peer review plays a huge role in project groups which is the evaluation phase. Using reports generated by the system, supervisors can evaluate each student from their progress and also the peer reviews that they gave by themselves

### **1.4.2 Specific Objectives**

In order to achieve the main objective there are specific objective to be completed. Specific objective that need to be attained is as followed.

1. Develop a component to auto generate project groups to create balanced group.

Project Group auto generation according to student’s given technological skills. Using Development technologies, preferred programming languages and previous projects. Using these parameters system will generate project groups non randomly and expecting to achieve maximum accuracy for fairness in project grouping. To achieve maximum accuracy students also should be honest about their skills. This system is using student skills and GPA to identify if the student is skilled or not. So depending on the GPA and skill data that they provide, student groups may change accordingly.

1. Develop a component to track project progress using GitHub and predict future Contribution

This solution is to track the project progress as each member’s contributions to the project. Using GitHub REST API, can track the project. We have to pass GitHub owner name, repository URL, repository name and the Personal Access Token. Using GitHub REST API, we can track each user’s commit contribution to the project and using that commits, we can predict future contribution. To the prediction process, we predict next week with last two weeks. System will generate prediction about future contributions by each student. Using this facility system can identify if there is a risk of leading to an incomplete project or not.

1. Develop a component to track client meetings

Client meeting platform convert voice to text and use emotional recognition to calculate the client satisfaction rate. The Meeting platform will allow the students to do video meetings with the client concerning the project. students can get outsource projects from system. To concerning the client after can create meetings, the system will do real-time voice-to-text converting and shows it so the student do not need to take the notes for client’s requirements. At the end of the meeting, can download the client speech as a requirement document. Also, the system can record the session in audio format.

1. Develop a component to peer review students to get an idea about each student.

Peer review mark generation based on monthly peer review reports. Peer review mark based on the average of monthly peer review marks which is gather by peer review form using question structure and feedback paragraph. These peer reviews can be used to evaluate each student. Honesty is a huge issue when it comes to project work. This peer review session will evaluate students’ performance by their own team members.

# 2. METHODOLOGY

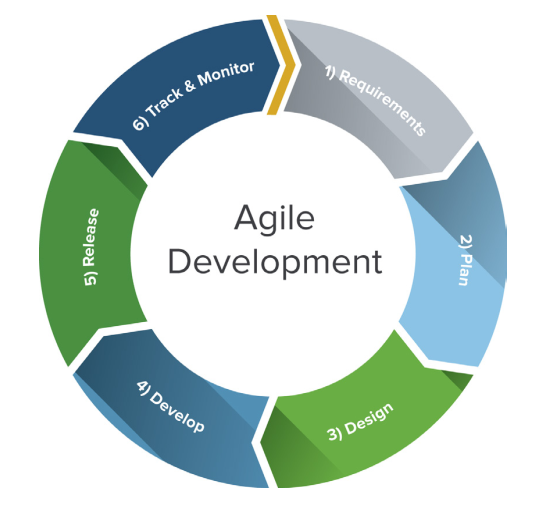
This section covers the procedures carried according to the software Development Life Cycle.

Figure 2: SDLC Life Cycle

For our solution, were using the agile methodology. The Internet economy has altered the rules of software engineering. Traditional development methodologies are too cumbersome to meet the rapidly changing requirements and short product cycles demanded by business. To meet these changing requirements, software developers have developed agile software development methodologies utilizing iterative development, prototyping, and templates [6]. In Each SDLC section will cover specific tasks used to complete the research components.

Since we’re using agile methodology, Development is an iterative process. Below list represents the flow of each iteration.

* Feasibility study
* Requirement gathering
* Designing
* Implementation
* Testing
* Maintenance

Given below is the work breakdown structure and we’ll explain how we completed each step with explanations.

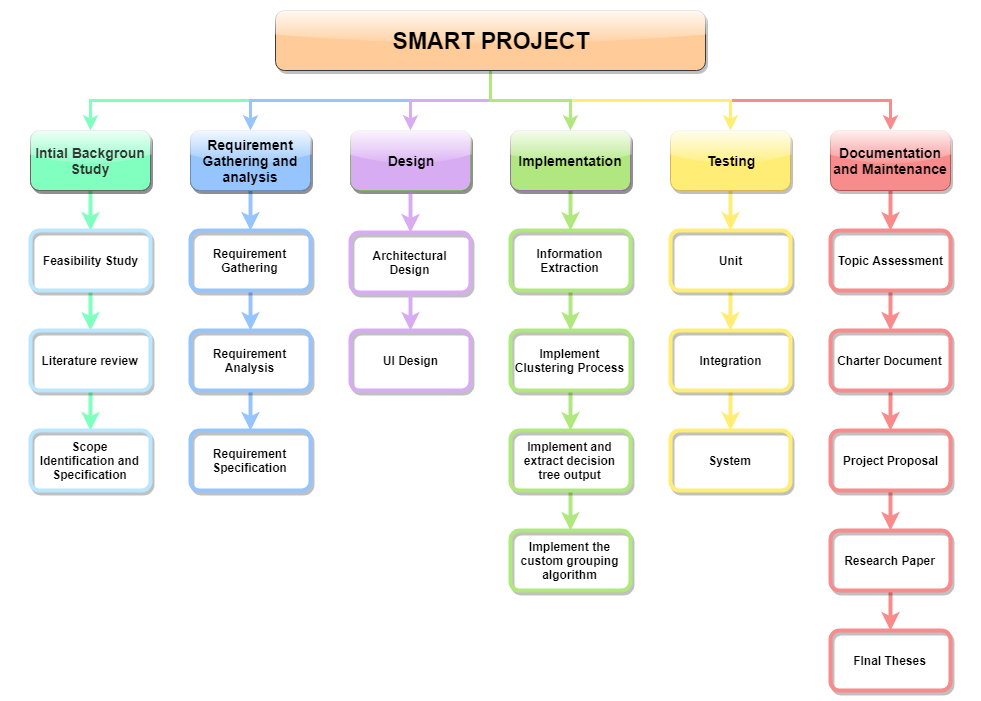


Figure 3: Breakdown Structure

## **2.1 Feasibility Study**

* **Schedule feasibility**

Proposed system should finished in the given time period and according to the research scope, should be able to finish within the time period and from the time period we can arrange additional time period will be used for testing before demonstrations. According to the present timeline we’re few days ahead of the schedule.

* **Economy feasibility**

Since the whole solution in deployed in a virtual environment, only for the server cost will be need to be taken care of. The solution can be also deployed in the university servers but for the security reasons, better to be deployed in a different server.

* **Technical Feasibility**
  + Machine Learning knowledge

To implement grouping algorithms, we need to have a high level understanding on machine learning knowledge to identify algorithms and the logic resides within the source.

* Basic knowledge in APIs and MySQL Server

Need to have a basic knowledge in API implementations. Since the API will be developed using java, we need to have basic knowledge in java.

* UI and UX

Need need to have atleast minimum knowledge in Frontend Frameworks since we’re using React JS as the frontend development.

## **2.2 Requirement Gathering and Analysis**

The first phase in the SDLC is the requirement gathering process. In the process we’ve gathered current issues with the project management process in the SLIIT ITP module. We’ve contacted the lecture in charge of that module to gather information about the module process and the current issues having within the management.

Survey Results

* Student Group creation

Usually there were conflicts between team members and the reason was the unbalanced group composition. Some groups having all the higher skill students and some of the groups having all the lower skilled students. So there is a need of a propper grouping mechanism for the project management.

* Unable to track project progress properly

Once the project started in universities, we realized that there was no consistance in project progress evaluation. Supervisors only evaluating the progress once a month or two months. Project evalutions will be much more reliable if these evaluations done in weekly basis. In order to do the process we’ve discussed a solution, which is GitHub contribution predictor and and also it records students project progress. Supervisors can track project progress from the beginning of a group project. Because this proposed system shows each member’s task progress and future prediction for each member. So the supervisor can get an idea about each member’s contribution for the group project from the beginnning of a project.

* Unable to track client meetings.

Specially in ITP module, we’ve realized that supervisors can not track client meetings. Students also doesn’t document meeting contents in order for later purposes, which is a bad practise. So we’ve discussed a solution to develop a client portal which supervisors can track meetings and also system automatically document meetings contents, such as vocal contents.

* Lack of peer review sessions.

We’ve discussed about previous evaluation phases of ITP module, and we’ve realized that evaluation phase is incomplete without proper peer review sessions. Peer review sessions help supervisors to identify free riders. Free riders are the students who doesn’t participating on group meetings and usually they doesn’t contribute to the project properly. So we’ve realized that we need a proper peer review process to evalutate students.

After the survey, we have come to a conclusion. To fix these issues we’ve proposed a project management system. This proposed project management system is consist of 4 major components. Which is group generation, GitHub contribution predictor, Client meeting tracker and finally the peer review sessions. Each component plays a major role in project management and also they are connected to each other.

## **2.3 Design**

The system architecture is shown in the figure 6.

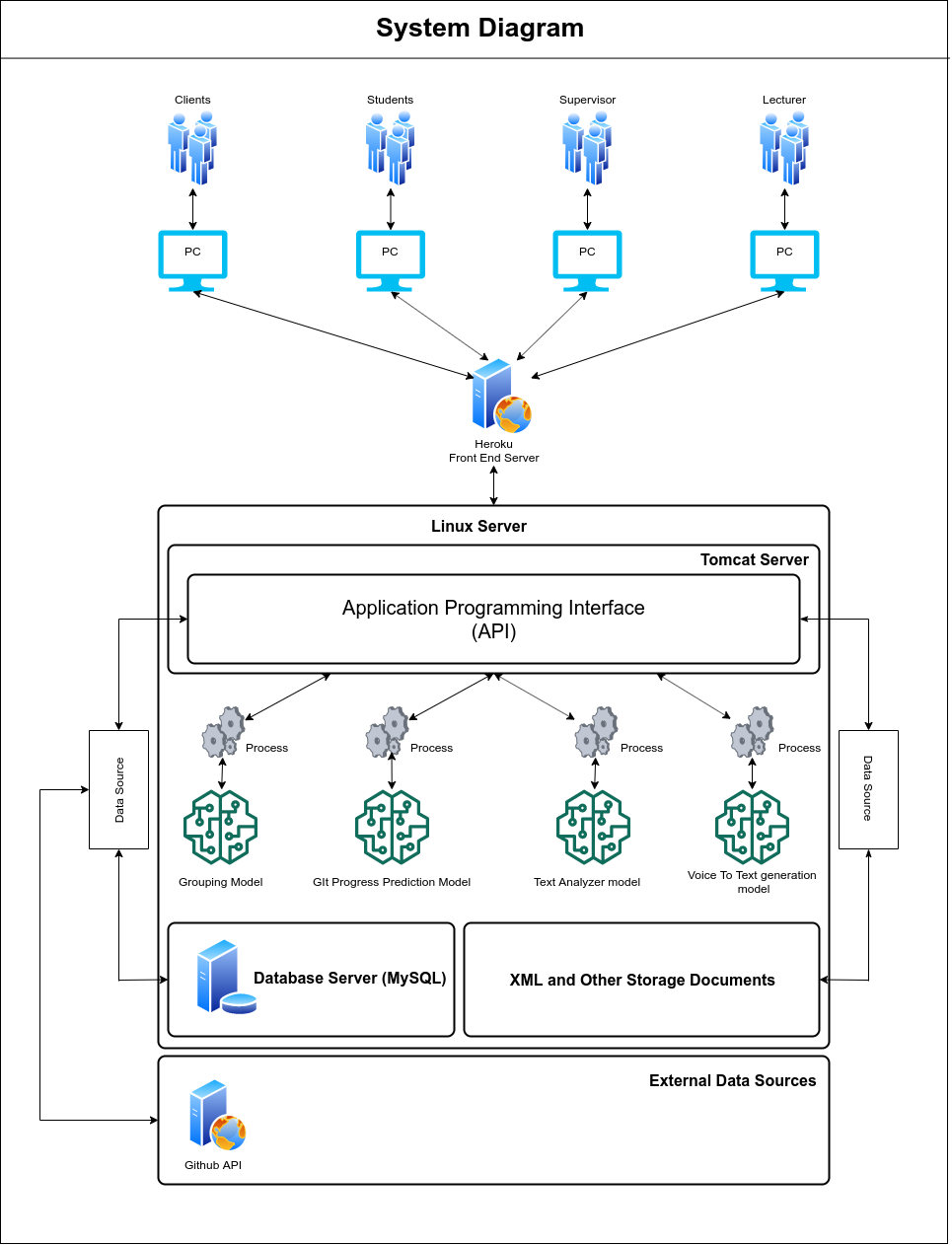


Figure 4: System Architecture Diagram

According to the system diagram as shown in Figure 4, We’re hosting the backend server in a Linux environment and inside the Linux server there are 4 components. The system is mainly based on client-server architecture. System will extract data from the system database and from the GitHub API. In addition to that, system maintain local file storage to manage necessary outputs from the system. Below list represents the 4 components that exists inside the system.

* API running using the exported quarkus API executable or user can host the executable inside an apache server.
* MySQL Server
* Trained Models for decision Makings
* XML and Other Storage Documents

Heroku Server used to deploy the Frontend solution to interact with users. Frontend solution will be communicating with the users and the API.

## **2.4 Implementation**

### **2.4.1 Group Generation**

#### **2.4.1.1 Information Extraction**

This group formation methodology considers 7 factors to increase the fairness in the group composition.

* Develop Technology
* Develop For
* Programming Languages
* Frontend Frameworks
* Backend Frameworks
* Database Management Systems
* Version Controlling

Using a student survey form and the student profile, we can extract student skills and other necessary data to proceed with the group formation process. But in order to start the process, first we need to acquire each required components from each and every student that enrolled in a module. Since the all required information can be extracted from the system itself, it is easy to proceed to next phases.

#### **2.4.1.2 Clustering and Decision Tree**

There are several strategies to group extracted student data. Widely used strategy is to create student groups randomly. But there are existing solutions that uses skill based analysis. For our system we’re using K-Means clustering algorithm to cluster student data into several groups. Basic concept is to create student lists considering different factors. Such as average GPA, development technology and programming languages.

Extracted student information firstly goes through the data clearing process and the generated student list will be go through the clustering process. Figure 5 represents the data cleaning code segment and Figure 6 represents the Clustering process code segment. Inside the clustering process students will be clustered into 3 clusters. Clustering process considers only the student GPA. Students will be clustered into High, Medium and Low GPA student groups. Then after the clustering process completed, system generates a csv file including the cluster number to identify the clusters in the next phase.



Figure 5: Data cleaning process code segment



Figure 6: Clustering process code segment

The issue of this process, system cannot identify a record if it is the High GPA record or a Medium GPA record. In order to identify the correct label for each record, we are using a Decision Tree algorithm.

After the clustering process, next phase is to label the clustered dataset. For this we’re using a decision tree algorithm. Figure 7 represents the code segment used to construct the decision tree.

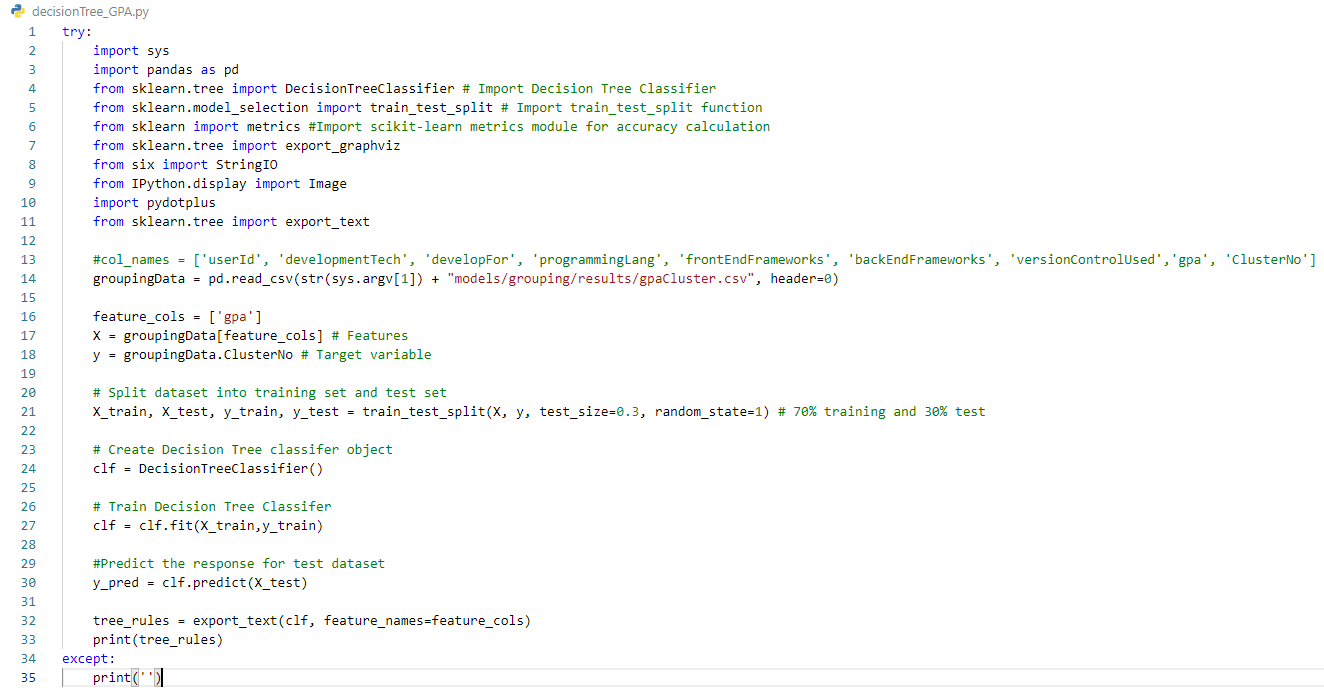


Figure 7: Decision Tree code segment

At this stage, all the necessary components of the grouping algorithm are acquired. Now we need to develop a custom algorithm that manage the group generation process. Since the group generation cannot be developed using previous group information, we need to develop an algorithm to decide which student is more suitable for a particular group. Next step is to build the custom algorithm to generate student groups.

#### **2.4.1.3 Custom Grouping Algorithm**

Using above mentioned 7 factors, we can divide students into different sections. For an instance we can provide sections as Desktop Applications, Web Applications, Mobile Applications. But the theory is not to just group them separately, but to mix all sections and create a fair group composition where it is not just random but the group composition is same for all generated groups.

##### **2.4.1.3.2 Group Formation Process**

In order to achieve fair and stable group composition, we create student groups with random students. This process is the majority uses to form student groups. Students will be randomly selected and system will generate student groups without considering any decision factor. This process helps the system to create a strong and fair group composition.

##### **2.4.1.3.3 Group Balancing Algorithm**

Group balancer will iterate through satisfied groups and unsatisfied groups. System select an unsatisfied group and it will exchange lower GPA students with High or medium GPA students from the satisfied group. After the exchange system will calculate the satisfaction rate of each group. If the satisfied group becomes an unsatisfied group, the process will be roll backed and then system will try the exact process with another satisfied group. Until each group achieve 100% satisfaction rate, the iterative process will be continued. But after a limited time on iterations, the loop will be ended to avoid any infinite loop occurrence.

#### **2.4.3.4 Group Balance Iterator Process**

In order to achieve maximum effect on the generated groups, we need to iterate this process using different marginal GPA values. In order to that, we’ve created another process which is the Group Balance Iterator. In this process system will try to identify what is the most suitable Average GPA for the conditional value. As we mentioned above, System uses the average GPA of initial student list. In this process, that conditional value will be reduced by 0.01 at each iteration. Once it reaches the suitable value, system returns the student groups with the satisfaction rate of 100%.

Once the system completes the Group Balance Iterator process, system can output the generated student groups as a list.

### **2.4.2 GitHub Contribution Predictor**

Implementation stage divides the part into several sections. Such as assigning Task or a Topic (That will be doing by the lecturer or supervisor. After created student groups, there is a lecturer will assign to the relevant group. Then the lecture will assign a task or a topic), Team leader has to publish the GitHub project URL to our system, get each contributions credentials with the Personal Access Token (API needs repository contributions’ credentials to configure), checking the code, commits and all other engagements related to each repository, analysis the evaluation of the repository (Commit charts, engagement charts, user engagements, some results related to code complexity and the quality of the code).

As we mentioned above, using GitHub REST API, can track the project. We have to pass the Owner name, repository name and the URL, and the Personal Access Token. After that we can get all the details regarding to that relevant project. We get all the above details when the leader creating a project environment in our proposed system. For the prediction part, we used Supervised Learning Algorithm to predict each member’s commit dataset. We don’t need a previous dataset or existing dataset to predict future contribution. We are using each project each member’s commits contribution as our dataset. We predict next week with the last two weeks.

Above mentioned objectives are the main objectives in this research part. For the sub objectives, Issue Tracking similar as stack overflow, notify students for incoming events and due date notifications, Performance report which represents individual contribution to the project by each team member.

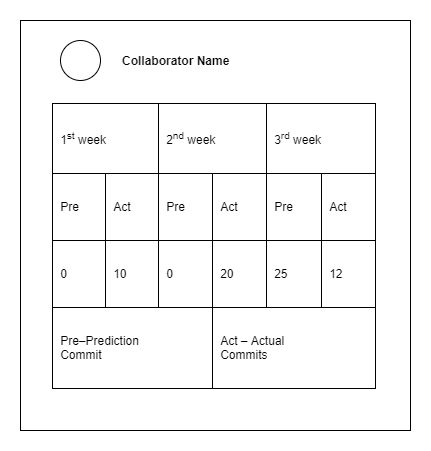


Figure 8: Prediction for a single user

In above figure, is the prediction component for a single user. As you can see there are several weeks and each week represents the actual commits and the prediction commits. As I mentioned in the methodology, next week will predict with the last two weeks. So prediction for 4th week will predict with 2nd and 3rd weeks actual commits.

After each section is completed it transfer to testing to verify and validation. we used React JS technology to develop the frontend part and Redux to store token in localhost (When some user log into the system there is a token to identify each uses. Redux will helps to store the token) and for the backend part we used python and JAVA. MySQL is our database. Visual Studio Code, Eclipse IDE and Xampp are the tools for relevant technologies.

As I mentioned in the above, Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. This uses labelled training data to learn the mapping function. We know,

* Classes as well as the labels.
* What attributes used in the model.
* Historical application details.
* We have the dataset.

The main reason to use Supervised Learning Algorithm is we have the data set. We collect the dataset as commits from previous weeks. As I mentioned above we need at least last two weeks commits to predict. Also we don’t need previous projects’ commits. We need only each project each commits per user. Because we predict each member’s future contribution. So we need only each project each member’s contribution as commits.

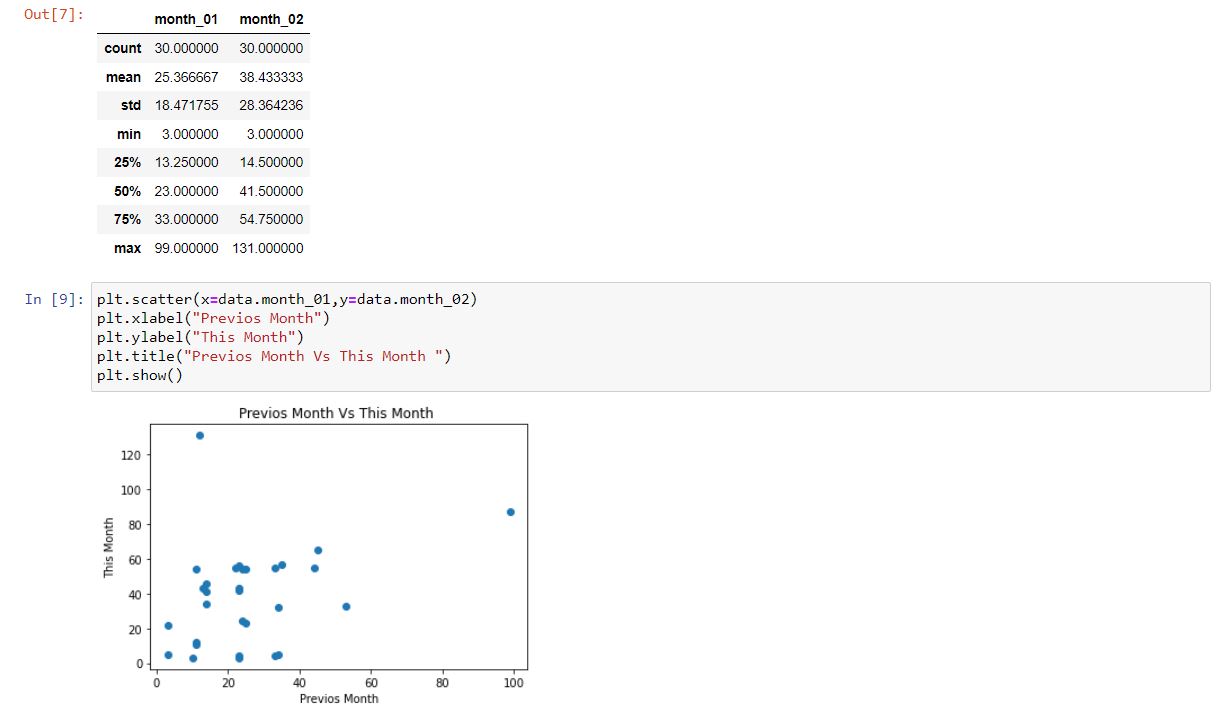


Figure 9: Describe of data and a chart with previous vs this month

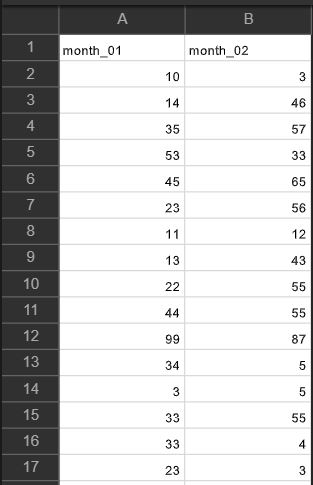


Figure 10: sample data set

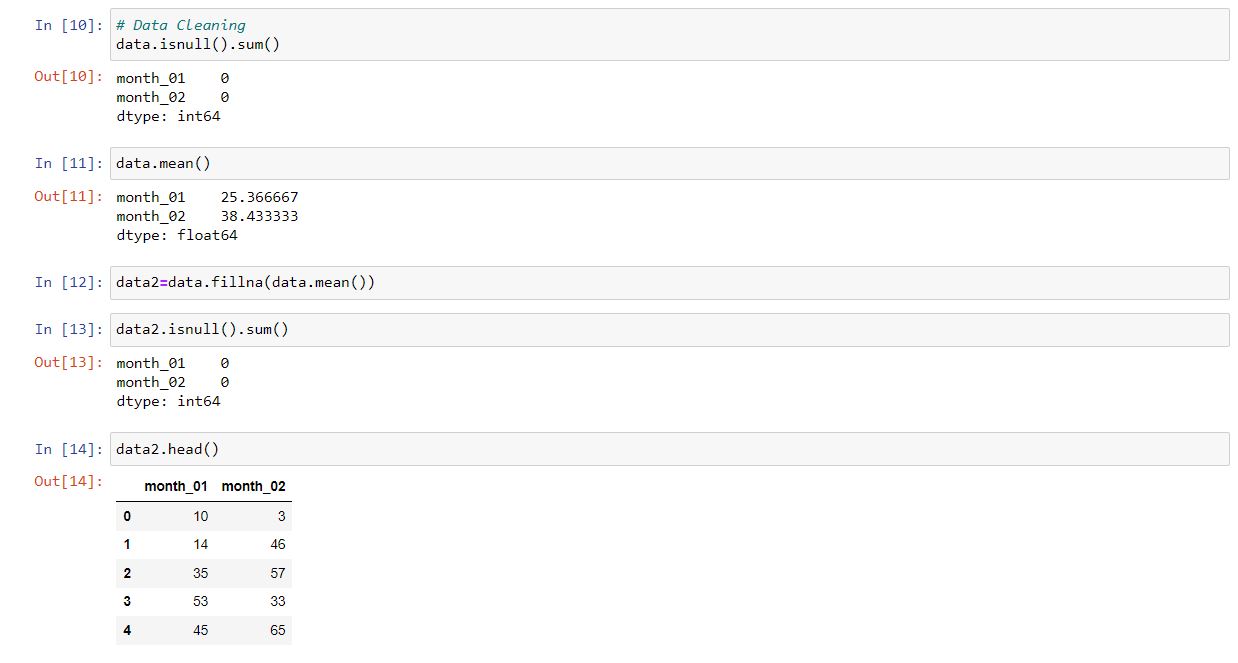


Figure 11: Data cleaning

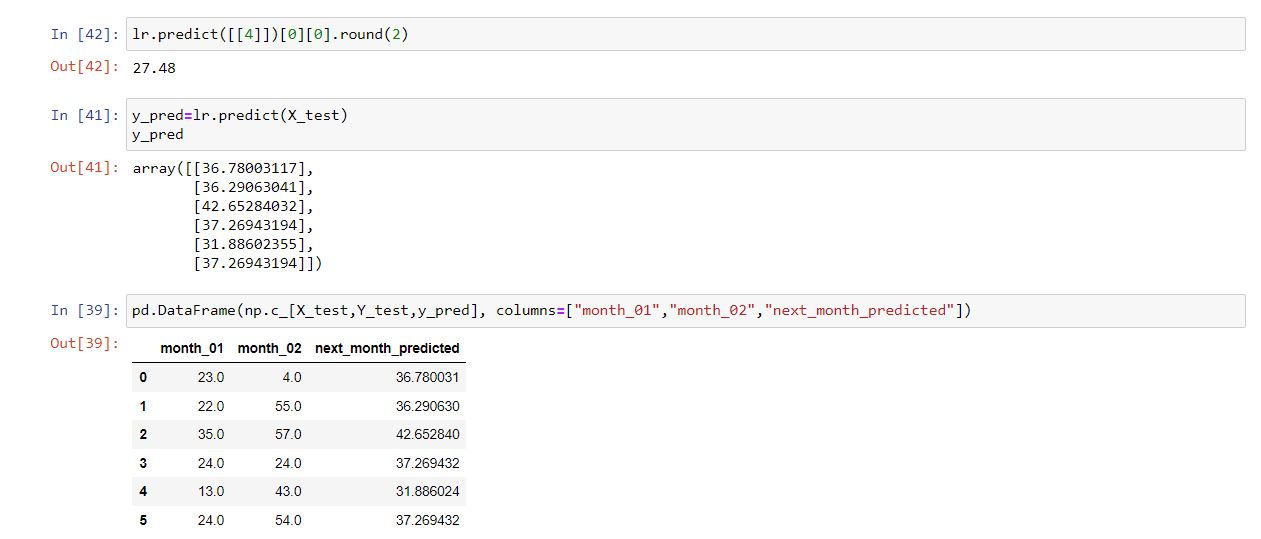


Figure 12: Select a model and train it

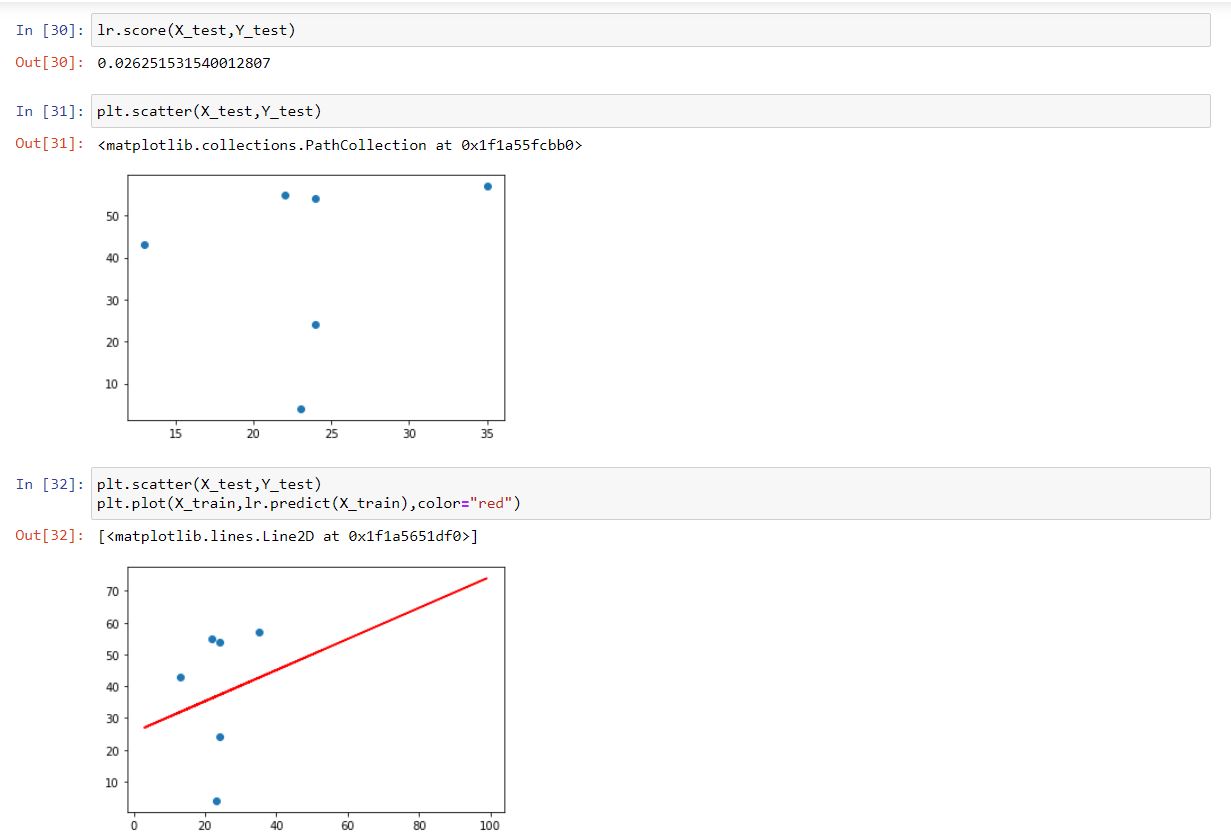


Figure 13: Fine Tuning the Model

### **2.4.3 Client Meeting tracker**

The implementation stage divides the part into several sections. Such as a client meeting platform, the emotional recognition system, Speech recognition system, checking and generating overall recognition report, classify client satisfaction rate, Client portal which provide facility to propose projects, suggest Functions for students which inserted by students

For the frontend part I am going to use React JS and for the backend part going to use python API. For the database we use MySQL and some java parts. The Visual Studio code and the Eclipse will be out tools to develop our system

In this system, we create a meeting platform that will allow the students to carry out video convene with the client to analyze the project. The client can have created the meeting and start the meeting in this part client can enter the meeting details and copy the id and can take a call, and continue the calling, after calling we can display the call receiving message and answer button. After the answer, the calling displays both videos in the meeting platform and enables the end call button. Figure 14 represents the meeting platform. And Figure 15 represents the meeting platform after Answering the call.

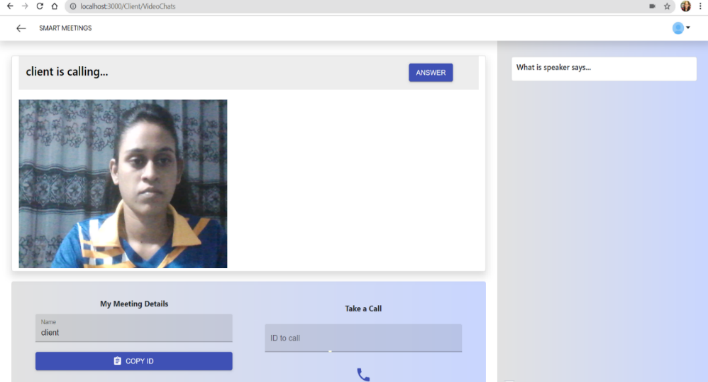


Figure 14: meeting platform

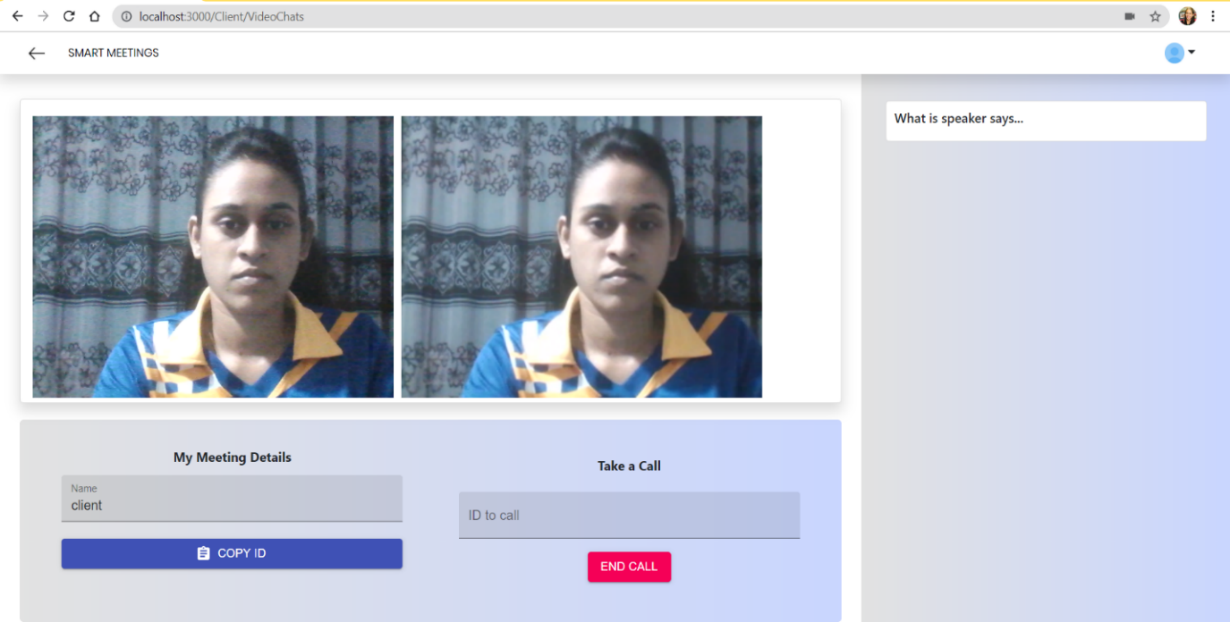
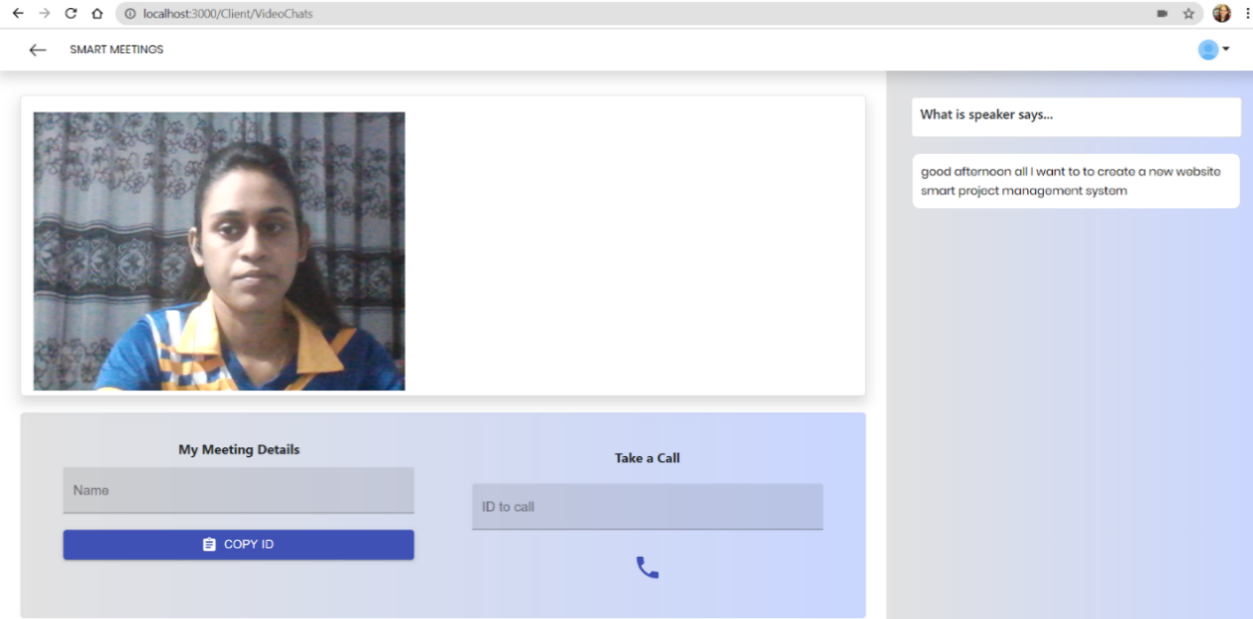
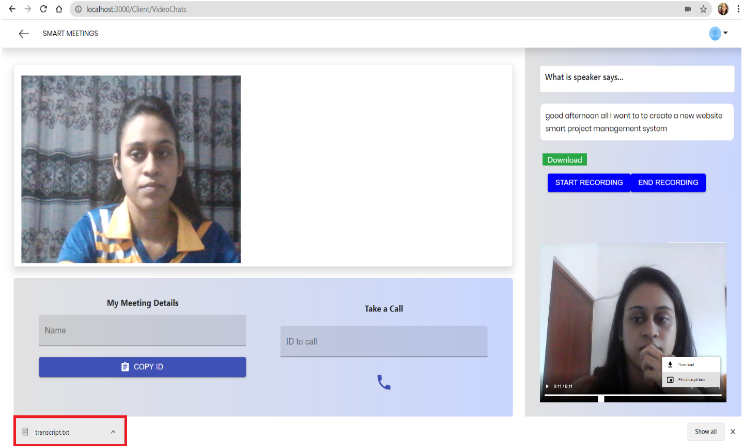


Figure 15: meeting platform after Answering the call

The system will do a real-time voice-to-text converting and shows them up on the screen.

Also, the system can record the meeting in audio format, which can be retrieved after the meeting. At last, the system will produce the report of the meeting which consists of text conversions done by the system and calculate the client's satisfactory rate using emotional detection from the text or emotion. Next, the system generates reports and allow to download those reports and video for future use. It will be very useful to students and clients. Since the students cannot remember all the requirements of the client, they can read the meeting report or listen to the audio recording to recall the client’s requirements. Figure 16 shows the live voice to text conversion and Figure 17 show the recorded video of the meeting.



Figure 16: shows the live voice to text conversion

##### Figure 17: show the recorded video of the meeting

Graphical user interface, application, Word

Description automatically generated

##### Figure 18: downloaded report

### **2.4.4 Peer Review**

The peer-review assessment component collects monthly student reviews to initiate the peer-review assessments with its main algorithm and other supportive algorithms. In the reviewing process project group member has to review all his project teammates monthly. review process time consumption will depend on the project team member count of the project team and project modules month allocations.

In the review process, all the team members have the fair ground with their team roles which means the project team leader will assess like all the other project team members of his project team. in the beginning instructions and guidelines to the proper review input will be given to the project team members of project modules through the web application and cloud space.

Month 1

Student 1

Student 2

Student 3

Month 2

Month 3

Figure 19: Student review path structure

Review

Textual

Numeric

Supportive Algorithms

Main Algorithms

Figure 20: Review types and usage

The peer-review process is collecting few types of reviews from each project member for its main algorithm and supportive algorithms. Overall the textual review is the prioritized one. Besides, those numerical reviews are also requested with the textual review to use in the supportive algorithms.

The review process is time-consuming and it should be. It is designed to be time-consuming on purpose to encourage an active learning environment. it doesn’t have a strict and fixed deadline. But the end of the month is required to be submitted as a responsibility of the project member.

A picture containing chart

Description automatically generatedNumeric review scores represent the positivity and the negativity of the project teammate's experience with a reviewed team member. Also, it represents the current situation with him as a project team member with the decision making in the project timeline which describes these scenarios in figure 21.

Figure 1: Numeric review scale description

This numeric review score will be compared with sentiment analysis paths positive sentiment score, negative sentiment score, and compound score to identify the role models of free-riding effect combined with group psychology which will be elaborate on results and discussion. The main algorithm design to divide its functionality after and within the pre-processing stage. Even after the separation, both algorithm paths will follow the basic functionality process of a machine learning environment as shown in Figure 22.

Text

Description automatically generated

Figure 22: Seven steps of machine learning approach

Diagram

Description automatically generated

Figure 2: Two major paths of the main algorithm

Before move on to the steps of each path’s machine learning stages of the algorithm, we will go through the main algorithm’s paths structure. The main algorithm divides into two paths to process the textual data achieve from the prior pre-processing stage of the algorithm. the first path leads the algorithm to the topic extraction component which has dominant topic coverage and coherence drop detection graphs. The second path leads the algorithm to sentiment score calculation to compare the numeric review scores get from the project group members.

#### **2.4.4.1 Data collection**

Data collection to the algorithm will input will be conducted through similar existing practices on project modules on the modules of the undergraduate project which is cloud space. beginning of the project module CSV template to review process will be published and share with project module students. After that students of the project module can maintain the template each month by providing the proper reviews to the peer-review component. Students of project modules are responsible for maintaining every month's reviews of every team member of the project group.

These review sources will go through processing steps in various stages to provide the optimal visualization to the algorithm to process the information on those stages. some of the stages will have to concatenate processes and some of the will have data frame reorder and redesign to provide more or less information to the algorithm.

Table 1: CSV template of the reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Review Id** | **Reviewee Id** | **Month** | **Review Score** | **Textual Review** |
| Student 1 | Student 2 | Month 1 |  |  |
| Student 1 | Student 2 | Month 2 |  |  |

**algorithm access range**

Project Team 001

**student access range**

Review Folder

Student 1

Student Reviews

Student1\_Student2.CSV

Student1\_Student3.CSV

Student 2

Student Reviews

Student2\_Student1.CSV

Student2\_Student3.CSV

Pre-processed Reviews

Figure 3: Access range comparison in the cloud space

As in figure 23 the maintenance of the review data, it will need a proper naming convention and access management to provide a positive feedback culture to the project module. properly named the CSV templates will lead to efficient data collection and productive review data maintenance in the cloud space for the algorithm's current and future usage. Also, the access range for the supervisor will be defined in the cloud space to maintain a positive feedback culture.

**algorithm access range**

Student1\_Student2.pdf

Student 1

Review Report

Project Team 001

**Supervisor access range**

Review Report

Review Folder

Student1\_Student3.pdf

Student 2

Figure 4: Access range comparison in the cloud space

#### **2.4.4.2 Data preparation**

In data, preparation algorithm will add and remove necessary rows and columns depend on its functionality of the stage to the above mention table 1 which is the basic input feedback collecting by peer-reviews in data collection. to visualize a theory or compare these reviews alongside with algorithm result from raw data won’t be enough that’s why there are data preparation stages include in most of the stages.

Numeric review Score

Positive Sentiment Score

Neutral Sentiment

Score

Score

Negative Sentiment

Score

Compound Sentiment

Score

Figure 5: Sentiment analysis data preparation

With this Sentiment score and Numeric review score comparison, we can demonstrate the team member's performance with both result-oriented and process-oriented indicators to achieve the expected learning outcomes.

The numeric review score is the only raw data material on above figure 25. all the other mentioned sentimental scores come from the sentiment analysis path outcomes. Those can be helper scores to the numeric review score most of the time. But if it's not hate speech or review bomb situation can be occurring on the review inputs to the algorithm.

#### **2.4.4.3 Data pre-processing**

In data-preprocessing there are several methodologies that exist in the research studies related to the sentiment analysis field with lots of approaches. in this research study, we are using regular expressions to go through the data-preprocessing stage of the algorithm.

**Diagram, schematic

Description automatically generated**

Figure 6: The text data preprocessing framework

#### **2.4.4.4 data preprocessing framework**

There are three major sections that follow in this data-preprocessing in the algorithm as shown in figure 27. some of them are light processes and some of them are heavy processes depend on the tasks and approaches with the algorithm. The noise removal is task-specific and tokenization and normalization is task-independent.

#### **2.4.4.5 Noise removal**

This algorithm's pre-processing pipeline goes through some light noise removal stages because we get data on specific CSV templates. If we get the data on HTML, XML, or JSON types of templates, we will need heavy noise removal with their headers, footers, and meta data.

Noise removal follows before any segmentation or tokenizing stage because of the contractions. The contraction will lead the tokenization process to split the without expanding which is not ideal for a sentimental analysis process. As shown in the figure 28 it can lead the word to invert the values and meanings.

Also putting this step after tokenization will generate the same result because of the segmentation already happens. We can take bigram functions to combined segmented negative words but it's highly unlikely to expect the same outcome we get from the noise removal process.

Contractions word

Standard Normalization

Tokenization

Advanced Normalization

Didn’t

Didnt

Did

nt

Did

Figure 7: Importance of noise removal process

#### **2.4.4.6 Normalization**

The normalization process comes after the noise removal process in the pre-processing pipeline. normalization will divide into two categories based on the functionality. The first one is the standard normalization process. as shown in the figure 26 it has a few sub-processes.

Remove new line characters

distracting single quotes

Remove punctuation

Lower casting Texts

Remove non-ASCII characters

Replace all integer occurrences

Figure 8: Standard Normalization

#### **2.4.4.7 Standard normalization**

Standard normalization is about setting up the fair ground for all the texts in the context. some of the steps taken in standard normalization can be re-use in upcoming stages or steps to validate or for the confirmation purposes such as removing punctuations as shown in figure 27.

Standard normalization

Actual Step

Remove punctuations (RE)

Tokenization

Validation Step

Remove punctuations (gensim)

Figure 30: Duplicate processes to validations

The validation process or the confirmation process will come as a build-in function on the some of the stages and steps depend on the resources usage. For example, in the figure 27 remove punctuations basically initiated in standard normalization as a regular expression function with the given data frames. But after that validation process comes with the tokenization stage with gensim utills which is a complete different approach.

#### **2.4.4.8 Advanced normalization**

Advanced normalization will come after the standard normalization which will set the fair ground for all the textual data. Advanced normalization again divides into three steps based on functionality. There stops word removal, stemming, and lemmatization.

#### **2.4.4.9 Stop word removal process**

Stop words can be identified as the common words used in textual data which has minimum impact on its overall meaning.in our research study, we are using nltk library to remove the stop words. Nltk is identified as The Natural Language Toolkit which is one of the well-recognized and recommended NLP libraries in the Python ecosystem which is used for tasks like tokenization, stemming, part of speech tagging and etc.

#### **2.4.4.10 Stemming process**

The stemming process is about removing affixes which including suffixes, prefixes, infixes, circumfixes from the given textual data to get the stem word of it.

Use

Stem word

Useful

Suffix

Stemming

-ful

Figure 31: Stemming -suffix removal

Suffix

Stemming

Stem word

ited

-Un

United

Figure 32: Stemming -prefix removal

Suffix

-s

passerby

Stem word

Stemming

passersby

Figure 33: Stemming -infix removal

Suffix

Stemming

Stem word

matur

-Im -ity

Immaturity

Figure 9: Stemming -circumfix removal

#### **2.4.4.11 Phrase detection - Bigram and Trigram models**

Bigram and trigram model building and generation is a helper function to the lemmatization process. Bigrams can be described as two words that score together in the textual data context. Trigrams are three words score frequently in the textual data context. in this research study, gensim phrases model is building bigrams and trigram for the next lemmatization process. There are two core arguments in the gensim phrases model which are min\_count and threshold. higher argument settings can be lead to low bigram and trigram count. After defining min\_count and threshold for the bigram and trigram models process will move on to the function calling for the models before the lemmatization process.

#### **2.4.4.12 Lemmatization process**

Lemmatization is bind to the stemming process but it is different because what is capturing is citation forms which are based on word’s lemma state. Lemma is referring to the head words which incudes in the dictionary.

Sometimes lexeme form which is the variation of the lemma or head word can differ from its lemma which can lead to change the impact it can put on the textual context.

But the sentimental value will be the same as the variation of the lemma. because of that lemma variations will not affect the sentimental value of the lemma even if it differs from the large margin to the head word chosen by the defined dictionary on the algorithm.

Also, the other processes like the stemming process or the tokenization process won't be affected to the word sentiment score based on if it comes prior to the lemmatization or after lemmatization because of the sentimental value won’t affect by the order of those processes.

Lexeme forms

matur

Lemma

Lemmatization

Breaking

Figure 10: Lemmatization process

#### **2.4.4.13 Topic Model**

##### **2.4.4.13.1 Pre-requisites**

To initiate any topic model it needs a proper input. in our research before move on to the topic model process we need two main inputs for our LDA topic model. There are the dictionary and corpus.

Corpus is a bag of words with a unique id and a frequency label created by the gensim library with the defined dictionary using the pre-processed data we given in earlier stages. when converting textual data to the bag of words library consider all the words are tokenized and normalized in the early stages. It can be existing in Unicode and utf8-encoded strings.no further pre-processing processes will be conducted after the tokenization, stemming, and lemmatizing stages. Dictionary is the module that encapsulates the mapping between pre-processed textual data with their integer ids. without a dictionary, corpus can’t be processed so corpus depends on the dictionary.

TOPIC

MODEL

BOW

Figure 11: LDA topic model process

##### **2.4.4.13.2 Model build**

Now the algorithm has the required input for the model but the algorithm needs a topic to count to the train the module. Depend on the model training environment we can decide the LDA model type. If the training environment has a low specification algorithm can run with gensim LDA model but if the training environment is capable of providing multiple cores algorithm can run with gensim LDA Multicore model.

Master process

Input

Queue

Input corpus iterator

Update

Model

process

Result

Queue

Chunks

Worker 1

Chunk

process

Figure 12: LDA Multicore process

In the multicore model master process consume one core of the training environment and all the worker consume given cores to their processes accordingly. in the master process chunks of the documents are divided and sent to the worker process and after worker processes are completed it master process collects them and updates the model process. LDA multicore module is relatively quick than LDA general model.

##### **2.4.4.13.3 Model training**

###### **2.4.4.13.3.1 Base model training**

There are several optional parameters that need to be set on the model when to initiate with the given topic count. There is the random state, update every, chunk size, passes, alpha, per word topics.

Chunk size is the number of documents used in the training chunk. chunk size is an integer and it's optional in the base model. With a multicore model, we can customize the chunk size for faster implementations. Passes also an optional integer type parameter which is the number of passes through the corpus in the training phase of the model. Update every is an integer and an optional parameter that defines the number of documents that will iterate on each update phase. If it is set to zero it will set to batch learning and if it set to one, it will set to online iterative learning.

The random state is optional integer parameter use in the reproducibility which can be an object or a seed to the model. Per word topics are the Boolean valued parameter. When its true model computes the list of topics and sorted it based on the likelihood of the topics for each word in the text.it will also multiply the phi values by word count.

Alpha is an optional float type parameter which most of the time is set to auto state.it has two other states named as symmetric and asymmetric. Alpha is a hyper parameter that will be tune in post stages. Alpha controls the mixture of the topic in the document. If we increase the alpha document will have a minimum mixture of topics. If we decreased the document will have a maximum number of topics.

###### **2.4.4.13.3.2 Model evaluation**

###### **2.4.4.13.3.3 Perplexity Score**

Shape

Description automatically generated with low confidencePerplexity score is a statically score of the model’s prediction on the given sample. It evaluates the model’s log-likelihood on the given test sample.to get the best outcome of the model predictions perplexity score has to be the lowest. Accurate probabilistic models have a high likelihood with low perplexity. But predictive likelihood scores and human judgments are least co-related, in some scenarios it can be even anti-correlated. [31]

Figure 13: Perplexity score

As figure 25 perplexities can be defined as the inverse probability of the given sample set which is normalized by the word count. [31]

###### **2.4.4.13.3.4 Coherence Score**

Topic coherence score monitors the list of topics generated by the model and the informative rate of the topics. There are multiple methods to calculate the coherence score such as C\_v, C\_p, C\_uci, C\_umass, C\_npmi, C\_a. in this research study we are we are using C-v and C\_umasss.

C\_v score or the coefficient of variance is based on single set segmentation of the top words with measures that have an indirect confirmation approach to normalized pointwise mutual information (NPMI) and cosine similarity. [32] [33]

**Text, letter

Description automatically generated**In this research study except for the coefficient of variance score, we are calculating the UMass score to validate the accuracy of the topic model.it measures the words only with properties with preceding and succeeding types and required an order list of words. In other words, its confirmation measures are based on document co-occurrences units and single preceding segmentation with logarithmic conditional probability. [33]

Figure 14: Umass calculation

Once each component finished its development phases, we’ve integrated the system using pyrunner script. Pyrunner script written to run python scripts in java backend programs. In order to run each component, first user need to install required python dependencies. Without the dependencies, Modules cannot run in the system. Once we install required dependencies, we’ve moved to deployment phase.

## **2.5 Software Solution Development Components**

### **2.5.1 Web Application Development**

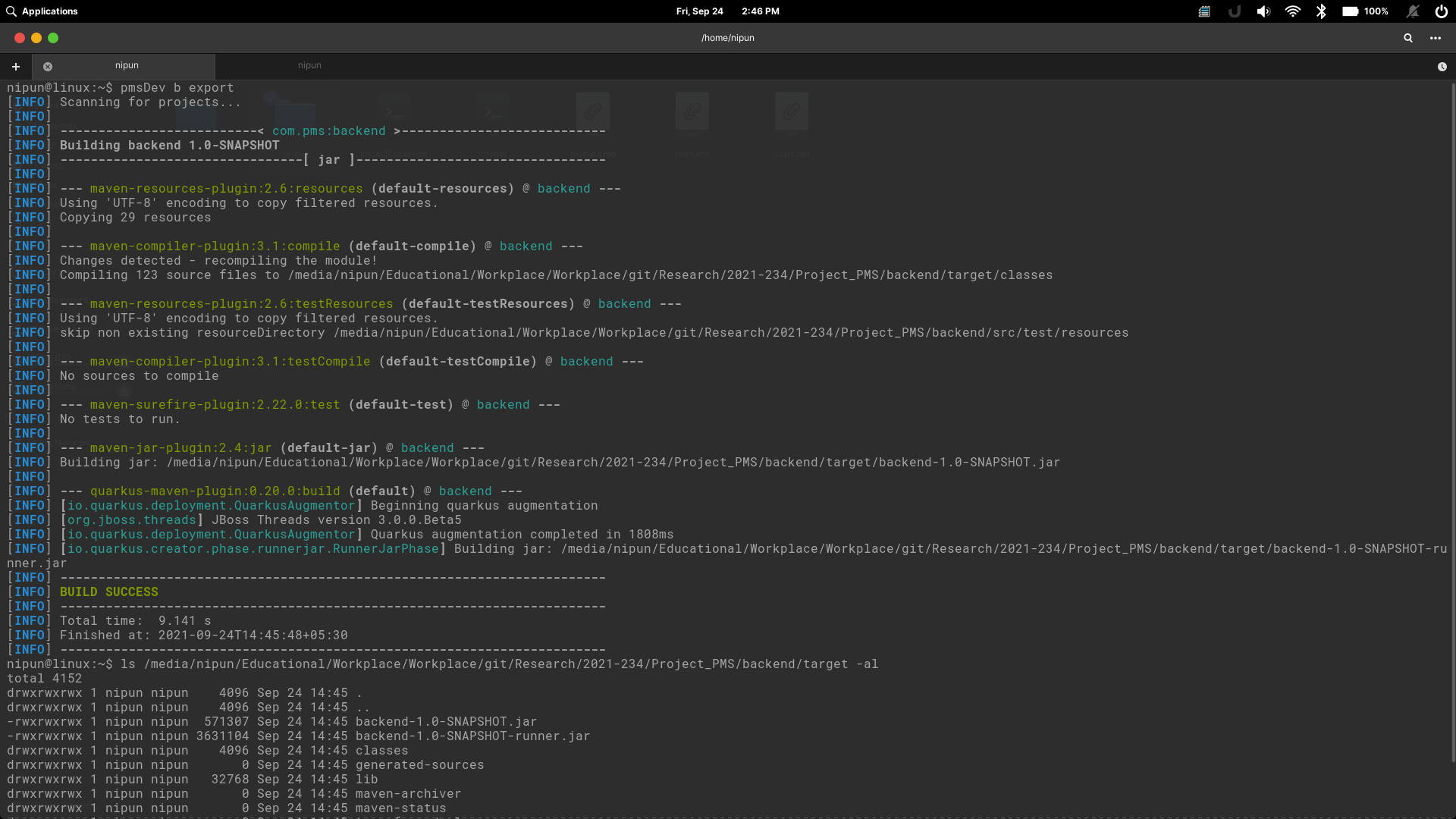
The end result of the project will be a web application to manage projects and the integrated capability to automate the grouping mechanism using student skills. We’ll be using React JS for the web application development. React is a JavaScript library for building user interfaces [7]. It is maintained by Facebook and a community of individual developers and companies.[[8]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-4)[[9]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-5)[[10]](https://en.wikipedia.org/wiki/React_(JavaScript_library)#cite_note-6). In development mode, we can simply run the web application using **npm run** command.

### **2.5.2 API development**

For the API development, we’re using java with maven quarkus[11]. Maven’s primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal, Maven deals with several areas of concern:

* Making the build process easy
* Providing a uniform build system
* Providing quality project information
* Encouraging better development practices

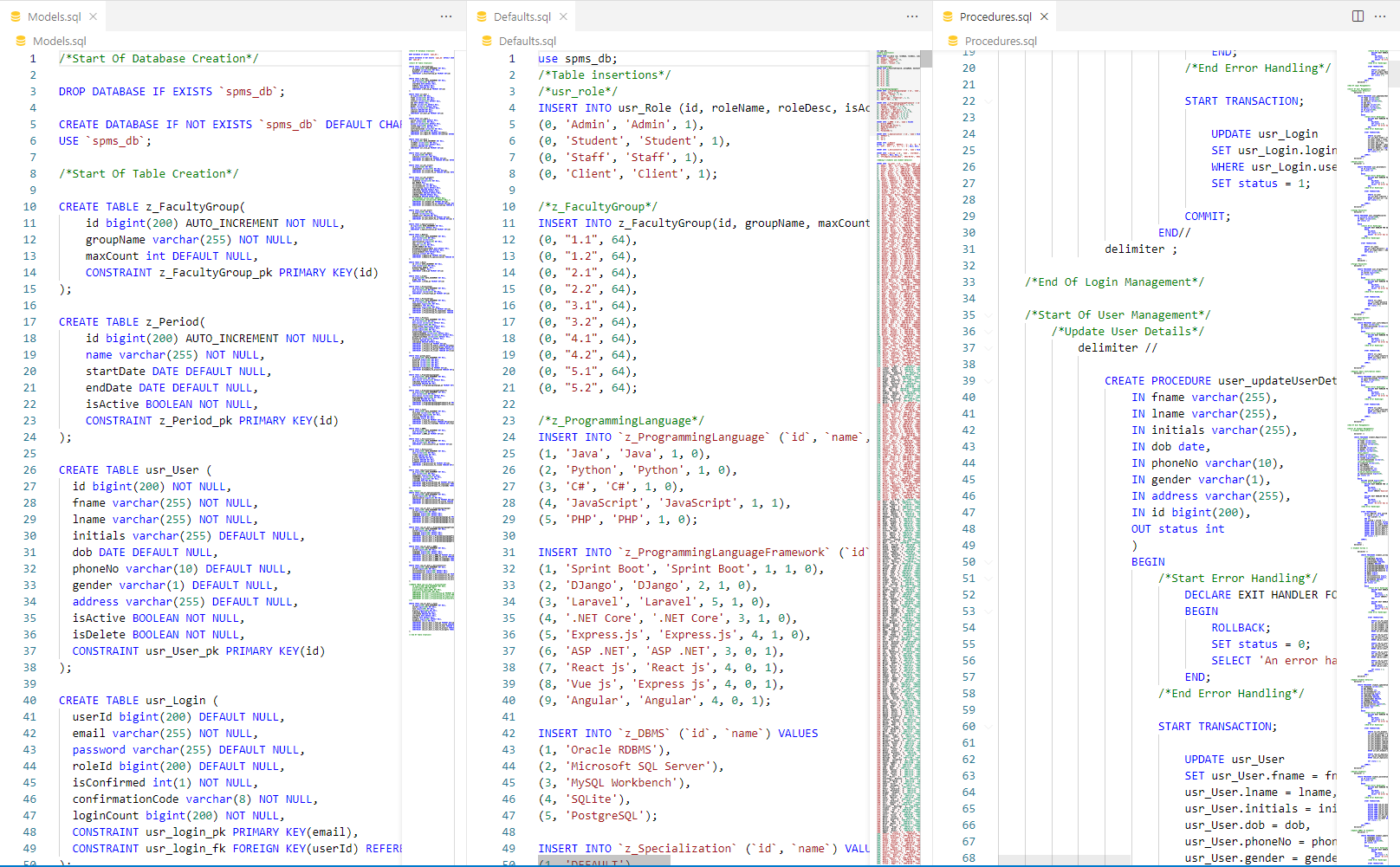
Maven aims to gather current principles for best practices development and make it easy to guide a project in that direction[12]. In development mode, we can execute the API by running **mvn compile quarkus:dev.** If users running the application inside a linux server, we’ve prepared a custom cammand script to execute application. In development mode, we can use **pmsDev b export** command to export the system to a jar file or else we can directly export and run the system by using **pmsDev b run** command. Figure 40 represents the application export process and Figure 18 represents the command to execute the exported jar file.



##### Figure 40: Export the system

### **2.5.3 Database Handling**

Using java JDBC driver makes it easy to manage MySQL database with the API. JDBC Driver is a software component that enables java application to interact with the database[13]. For the database querying and configuring database structures, we’ll be using MySQL workbench. This database structure will contain Tables, Procedures and functions to reduce the code repetition. In addition to that, we’re maintaining different files for different purposes. Such as, Database tables(models), Default values for tables, Procedures for the functionalities. We’ve created procedures only for write and update operations. Figure 41 represents the db script code segments for the database.



##### Figure 41: Database scripts

### **2.5.4 Testing**

**PostMan**

For the API testing we’re using Postman. Postman is a collaboration platform for API development. Postman's features simplify each step of building an API and streamline collaboration so the user can create better APIs faster[14].

In order to achieve maximum stabality in the system, we’ve continued the unit testing process until we integrate the system. Below tables are the test results of each unit testing phases.

Table 2: Test case 01

|  |  |
| --- | --- |
| Test case ID | 1 |
| Test Case description | Verify if a user will be able to login with a valid email and valid password. |
| Pre- condition | Get the URL and connect the System |
| Test procedure | Get the URL and connect the System and Input the data |
| Test Input Data | Email and Password |
| Expected Output | User can logging the system successfully |
| Actual Output | User can logging the system successfully |
| Result | Pass |

Table 3: Test Case 02

|  |  |
| --- | --- |
| Test case ID | 2 |
| Test Case description | Verify if a user cannot login with an invalid username and an invalid password |
| Pre- condition | Get the URL and connect the System |
| Test procedure | Input the invalid data |
| Test Input Data | Invalid user name or Invalid password |
| Expected Output | System should display the error message Can’t logging the system |
| Actual Output | System should display the error message Can’t logging the system |
| Result | Pass |

Table 4: Test case 03

|  |  |
| --- | --- |
| Test case ID | 3 |
| Test Case description | Once user logged to the system increase the login count. |
| Pre- condition | User must login to the system. |
| Test procedure | Once user authenticated, increase the values of loginCount in the database. |
| Test Input Data | Email and Password |
| Expected Output | If the previous loginCount values is 5, next values should be 6. |
| Actual Output | System recorded loginCount as 6. |
| Result | Pass |

Table 5: Test case 04

|  |  |
| --- | --- |
| Test case ID | 4 |
| Test Case description | If someone registered to the system, a confirmation code should sent to them via the given email. |
| Pre- condition | User must give required details for the registration process. |
| Test procedure | Once the system verify the given data, system generates a random confirmation code and send it to the user via email. |
| Test Input Data | User details |
| Expected Output | System redirect user to the login page, but system shows “Verify the login first. Check your emails” when he tries to login. |
| Actual Output | System blocks the login process and shows Verify the login first. |

Table 6: Test case 05

|  |  |
| --- | --- |
| Test case ID | 5 |
| Test Case description | Account verification via a verification link. |
| Pre- condition | User must locate the email and click the confirmation link. |
| Test procedure | Once the user click the link, the verification code will be sent to the server and the server will check the validity of the verification code. |
| Test Input Data | Verification Code. |
| Expected Output | Since the verification code is embedded with the link, Human errors should be zero and system should verify the login once the user click the link. System will show user “User account is activated” |
| Actual Output | Once the user click the link, system shows user account activated. |
| Result | Pass |

Table 7: Test Case 06

|  |  |
| --- | --- |
| Test case ID | 6 |
| Test Case description | Verify if the user can create a new project environment. |
| Pre- condition | Go to add new project page using student profile sidebar |
| Test procedure | Input data to create new project |
| Test Input Data | Fill all the data |
| Expected Output | User create a project and system will display a toast message as a success message |
| Actual Output | System should display the success message |
| Result | Pass |

Table 8: Test Case 07

|  |  |
| --- | --- |
| Test case ID | 7 |
| Test Case description | Verify if the user tries to create a new project without required fields. |
| Pre- condition | Login into the system as a student |
| Test procedure | Miss GitHub filling fields. |
| Test Input Data | Fill only basic details. Not the GitHub data fields. |
| Expected Output | System should display the error message |
| Actual Output | System should display the error message |
| Result | Pass |

Table 9: Test Case 08

|  |  |
| --- | --- |
| Test case ID | 8 |
| Test Case description | Get all Project that related to logged in user. |
| Pre- condition | Login into the system as a student |
| Test procedure | Go to all projects page |
| Test Input Data | Pass the user ID |
| Expected Output | Get all Projects for the logged in user. |
| Actual Output | System will display all the projects inside the all project screen. |
| Result | Pass |

Table 10: Test Case 09

|  |  |
| --- | --- |
| Test case ID | 9 |
| Test Case description | Get a Single Project |
| Pre- condition | Login into the system as a student |
| Test procedure | Click one project in the all projects screen |
| Test Input Data | Pass the project ID |
| Expected Output | Get single project’s all data. |
| Actual Output | Single project screen will display with the project data. |
| Result | Pass |

Table 11: Test Case 10

|  |  |
| --- | --- |
| Test case ID | 10 |
| Test Case description | Display prediction process in the single project page |
| Pre- condition | Login into the system as a student |
| Test procedure | Click prediction tab that is inside the single project page. |
| Test Input Data | Pass commits per a collaboration |
| Expected Output | Prediction table will display with actual commits and the prediction commits for next week |
| Actual Output | Prediction table |
| Result | Pass |

Table 12: Test case 11

|  |  |
| --- | --- |
| Test case ID | 11 |
| Test Case description | Student participate to the student survey, and system verify the user provided data. |
| Pre- condition | User must login to the system as a new user. If user is not a new user, the student can navigate to the student survey via the student profile in the system. |
| Test procedure | Once the student navigate into the student survey, student must fill every field before submitting the form. |
| Test Input Data | Student educational details and skill data. |
| Expected Output | If the student leave a field as empty when the student submitting the form, system shows a message “This field is empty” with the name of the field. |
| Actual Output | When student leave the programming languages field as empty, system showed as “This filed is empty: programming languages”. |
| Result | Pass |

Table 13: Test case 12

|  |  |
| --- | --- |
| Test case ID | 12 |
| Test Case description | When user try to create student groups, user must select a module. |
| Pre- condition | User must navigate to Group generation UI. |
| Test procedure | To initialize the group generation process, user should select a module from the dropdown. |
| Test Input Data | Module details. |
| Expected Output | If user select a module before running the group generation process, system will generate student groups without any issue. But if user doesn’t select a module before group generation, system shows “Please select a module.” |
| Actual Output | user selected the module before running the group generation process and system generated student groups without any issue. In second time, user didn’t select a module before group generation. system showed “Please select a module.” |
| Result | Pass |

Table 14: Test case 13

|  |  |
| --- | --- |
| Test case ID | 13 |
| Test Case description | When user try to create student groups, user must enter number of students per group. |
| Pre- condition | User must navigate to Group generation UI. |
| Test procedure | To initialize the group generation process, user should enter a number as the student count per group. |
| Test Input Data | Student count per group as a integer. |
| Expected Output | If user enter a number before running the group generation process, system will generate student groups without any issue. But if user doesn’t enter a number before group generation, system shows “Please enter a number for student count”. If user enter a negative value as a student count, then the system shows “Please enter a positive values as the student count”. |
| Actual Output | user entered 5 as student count before running the group generation process and system generated student groups without any issue. In second time, user didn’t enter a student count before group generation. system showed “Please enter a number for student count.”. When user entered a negative number as the student count, system showed “Please enter a positive values as the student count”. |
| Result | Pass |

Table 15: Test case 14

|  |  |
| --- | --- |
| Test case ID | 14 |
| Test Case description | Verify the password field is either visible as asterisk or bullet signs. |
| Pre- condition | Get the URL and connect the System |
| Test procedure | Get the URL and connect the System  Enter the user name and password |
| Test Input Data | Username and Password |
| Expected Result | The system should display the password on asterisk or bullet signs. |
| Actual Result | The system should display the password on asterisk or bullet signs. |
| Result | Pass |

Table 16: Test case 15

|  |  |
| --- | --- |
| Test case ID | 15 |
| Test Case description | Verify the Meeting button is working |
| Pre- condition | Log the system as a client |
| Test procedure | Get the URL and connect the System  Enter the correct user name and password  Click the meeting button of dashboard |
| Test Input Data | Click the button |
| Expected Result | System should open the SMART MEETINGS video plat form |
| Actual Result | System should open the SMART MEETINGS video plat form |
| Result | Pass |

Table 17: Test case 16

|  |  |
| --- | --- |
| Test case ID | 16 |
| Test Case description | Verify the meeting details (Name, Call ID) |
| Pre- condition | Log the system as a client |
| Test procedure | Get the URL and connect the System  Enter the correct user name and password  Click the meeting button of dashboard  Enter the name of the client  Copy the Meeting ID and enter it ID to call field |
| Test Input Data | Name , Meting ID |
| Expected Result | System can create the meeting |
| Actual Result | System can create the meeting |
| Result | Pass |

Table 18: Test case 17

|  |  |
| --- | --- |
| Test case ID | 17 |
| Test Case description | Verify the call receiving message and answer button is display on top of the screen |
| Pre- condition | Logging the system  Create the meeting and get the call |
| Test procedure | Enter the meeting ID  Click the Calling icon button |
| Test Input Data | Meeting ID |
| Expected Result | System should display the call receiving message and answer button is on top of the screen |
| Actual Result | System should display the call receiving message and answer button is on top of the screen |
| Result | Pass |

Table 19: Test case 18

|  |  |
| --- | --- |
| Test case ID | 18 |
| Test Case description | Verify the real-time voice-to-text converting text part display on the screen |
| Pre- condition | Create the meeting |
| Test procedure | Create the meeting  Strat the conversations |
| Test Input Data | Voice |
| Expected Result | System should display the real time voice converting text part on the screen |
| Actual Result | System should display the real time voice converting text part on the screen |
| Result | Pass |

Table 20: Test case 19

|  |  |
| --- | --- |
| Test case ID | 19 |
| Test Case description | Verify report download button is working properly |
| Pre- condition | Create the meeting |
| Test procedure | Create the meeting  Strat the conversations  Click the download button |
| Test Input Data | Voice |
| Expected Result | Can download the report |
| Actual Result | Can download the report |
| Result | Pass |

#### **Sensitivity Tests and hyper parameter tuning**

In this section, we will go through the hyper-parameter tuning functions. As described in the methodology hyper parameter tuning is conducted by series of sensitivity tests. With sensitivity tests, we are setting up the optimal alpha value, beta value, coherence, and the topic count. This figure 42 is belongs to the sample of the first validation set which is 75% of the corpus. topics count and alpha values are stable to two and 0.01 but both alpha value and coherence values are incrementing.



Figure 42: Sensitivity test results part 1

In this figure 43, its alpha value change to symmetric type from the auto but topics count is still stable. Corpus is the same as the previous validation set. But we can see the coherence drop with beta increment.

Figure 43: Sensitivity test results part 2

In this figure 44, we can see the alpha value again shift from symmetric type to asymmetric type. Still, the same topics count with the same validation set. The beta increment is the same as the previous. But we can see the coherence is unstable in this asymmetric alpha status. it has two coherence drops.



Figure 44: Sensitivity test results part 3

Now we move on to the next validation set which is the full corpus sample. in this figure no, we can see symmetric alpha values with beta increment and coherence drop to the stable topic count.



Figure 45: Sensitivity test results part 4

This second iteration of the full corpus sample has shifted its alpha values to asymmetric from the symmetric type and we can see a significant coherence drop to a beta increment and a stable topic count.



Figure 46: Sensitivity test results part 5

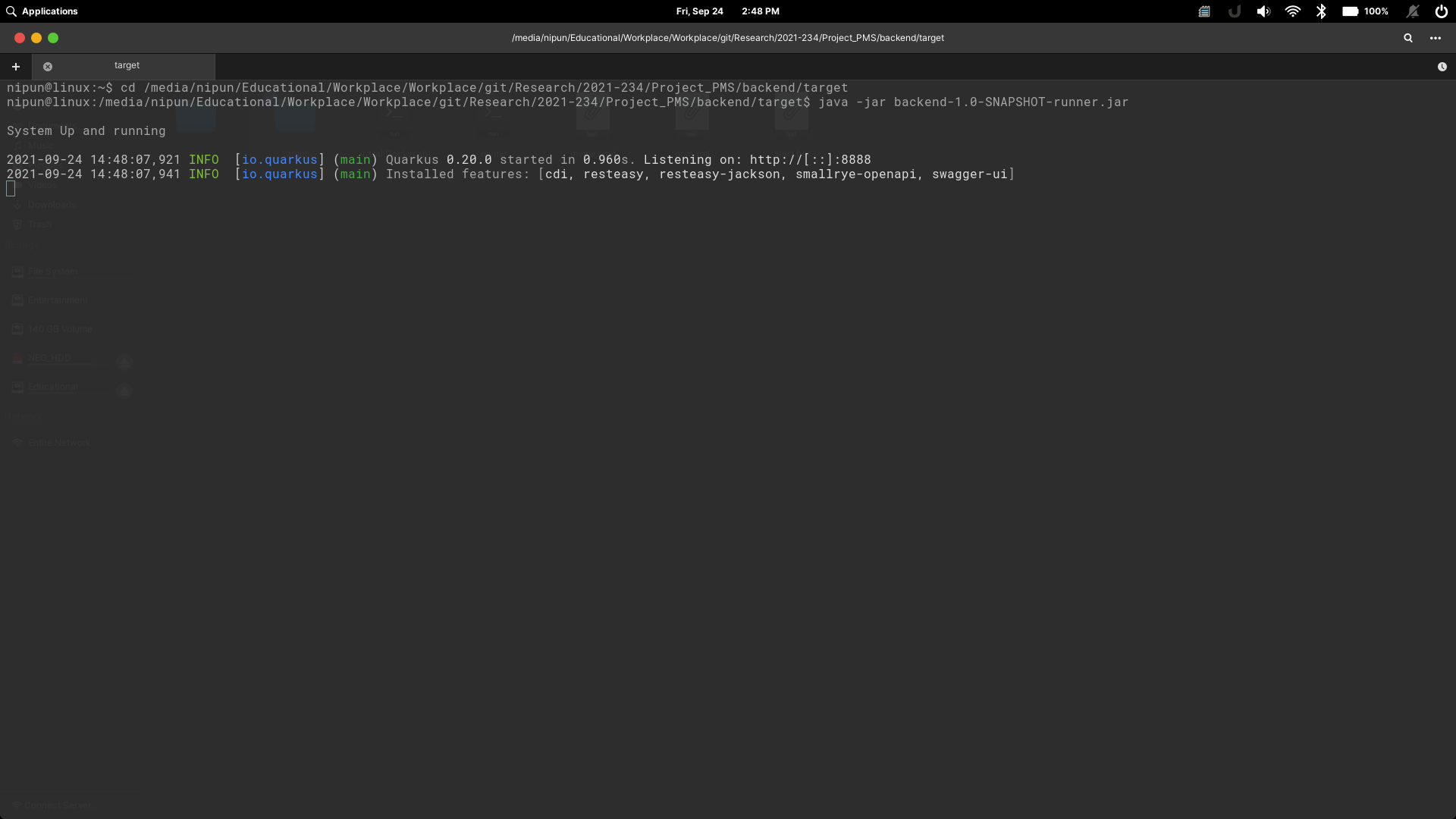
In the previous iteration, we see topic counts stable with the symmetric and asymmetric count. But here its shifts to default alpha values with topic increment by one level and still beta is incrementing. In the full corpus sample, we can see significant coherence drops more often than in the first validation set.

Figure 47: Sensitivity test results part 6

### **2.5.5 Deployment**

**AWS**

AWS has significantly more services, and more features within those services, than any other cloud provider–from infrastructure technologies like compute, storage, and databases–to emerging technologies, such as machine learning and artificial intelligence, data lakes and analytics, and Internet of Things. This makes it faster, easier, and more cost effective [15].



##### Figure 48: Execute the system

**Heroku**

Heroku is a cloud platform as a service supporting several programming languages. Heroku is used for container-based cloud Platform as a Service (PaaS). As we mentioned in the system overview, we deploy our frontend part in Heroku cloud platform. To do that we create a heroku account and initialie our project as a heroku GitHub Repository. Then we push our files to heroku repository. After that Heroku will automatically deploy our project to their servers. Finally, they are providing a URL to access our frontend.

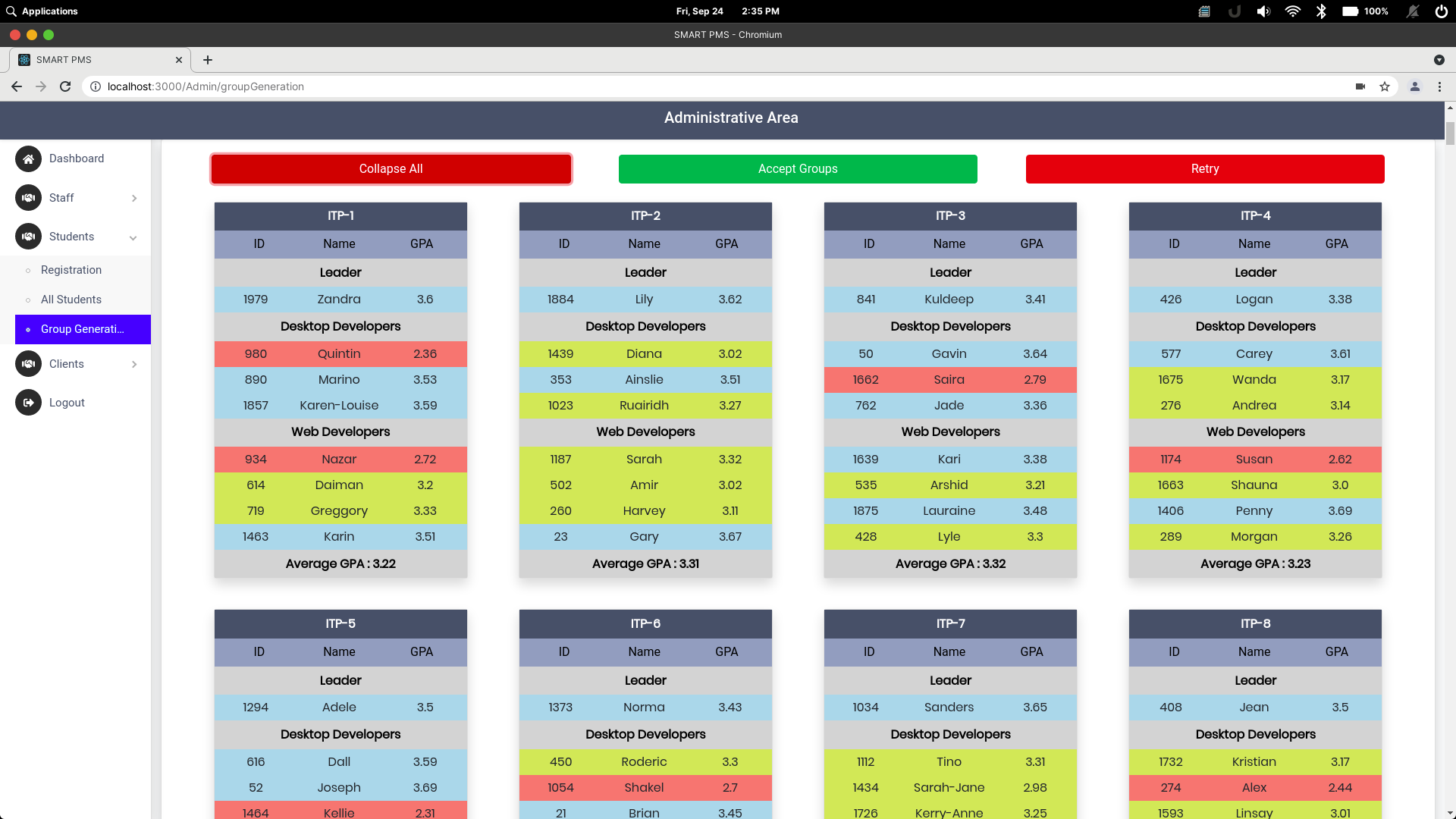
# 3. RESULTS AND DISCUSSION

## **3.1 Results**

### **3.1.1 Group Generation**

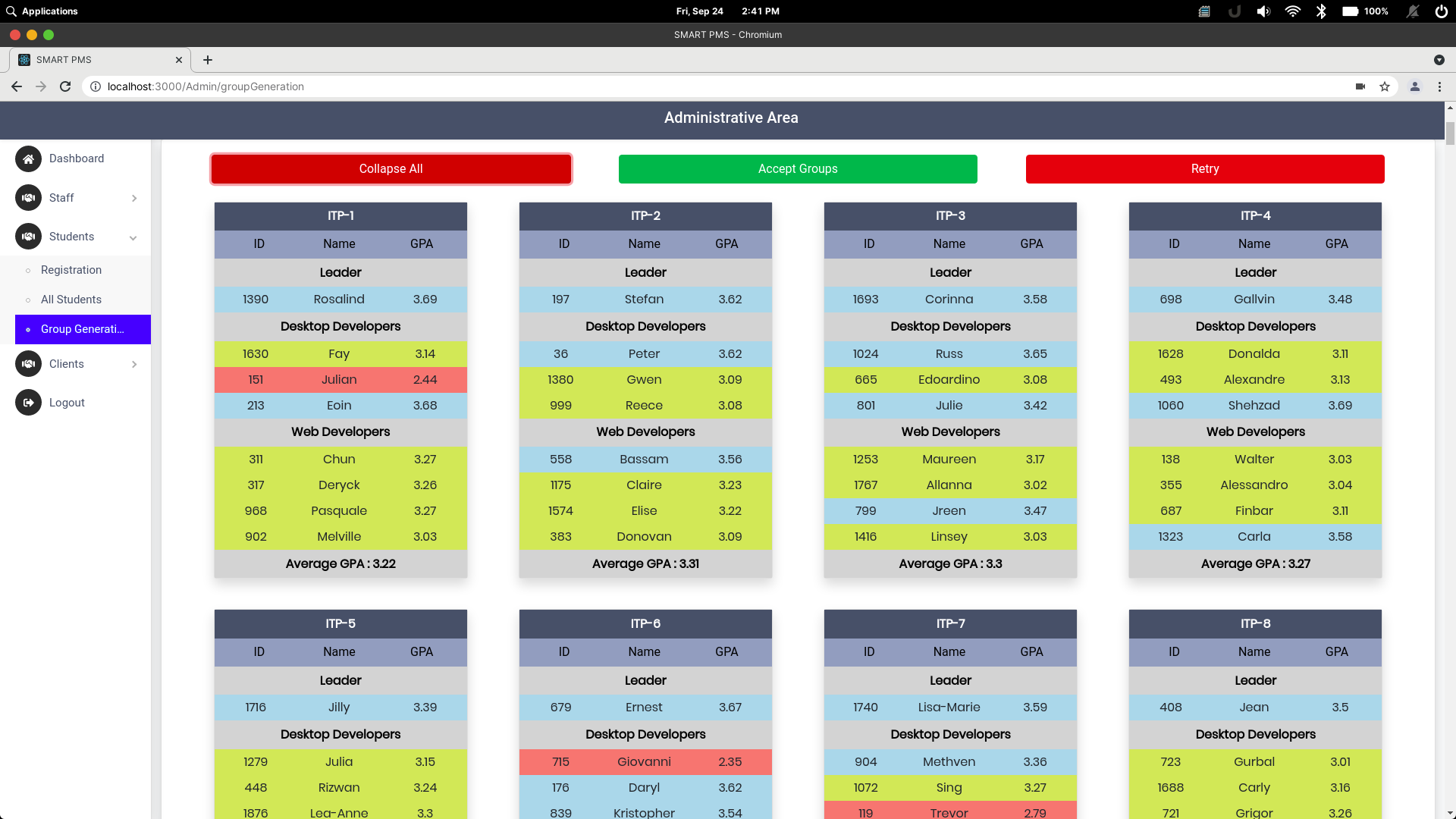
Once the system aquire student details and their skills, system was able to generate a student list with their skills. Complete system worked efficiently and interectively with student details. After aquiring student details, next step was to generate student groups with the student details. User need to navigate to the group generation UI in the system. Once the user enteres to the group generation UI, user need to select the module from the drop down and also the number of students per group. Then user can start the group generation process. It takes few minutes to generate groups since this is a iterative process. Once group generation is finished system shows the generated groups. User can decide from there, whether to continue with the generated groups or regenerate new groups.

Once user select Accept, system will store group data inside the database. After that students can login to the system and check for their student group and contact their team mates via client portal or using their emails. Figure 49 represents the final outputs from the group generation UI.



##### Figure 49: First group generation Output

In each iteration system will generate different outputs. Reason is to that we’re trying to create student groups more diverse than any manual strategy. So the process output can be much more different than the previous iteration. Since the iteration prcess checking average GPA of all generated groups in order to achieve the highest average GPA to increase the effectiveness of the system. Figure 50 represents the second iteration and group composition of both outputs are different from the other.



##### Figure 50: Second group generation output

### **3.1.2 GitHub prediction results**

This research part implementation worked collectively and interactively. GitHub prediction was able to predict the future contribution to the project relatively to each group member. First of all, Process should start with the grouping process. This is another research from my team. After the group generation process, the next step is to manage projects and project repositories. In order to do that, system uses GitHub integration using GitHub API. By retrieving repository data, System can generate GitHub contribution predictions.



##### Figure 51: final result

Above figure represent a final result for a predicted dataset. This came from after the fine tuning the model.

Table 21: Prediction Overview for User X

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1st week | | 2nd week | | 3rd week | |
| Pre | Act | Pre | Act | Pre | Act |
| 0 | 10 | 0 | 20 | 25 | 12 |
| Pre–Prediction Commit | | | Act – Actual Commits | | |

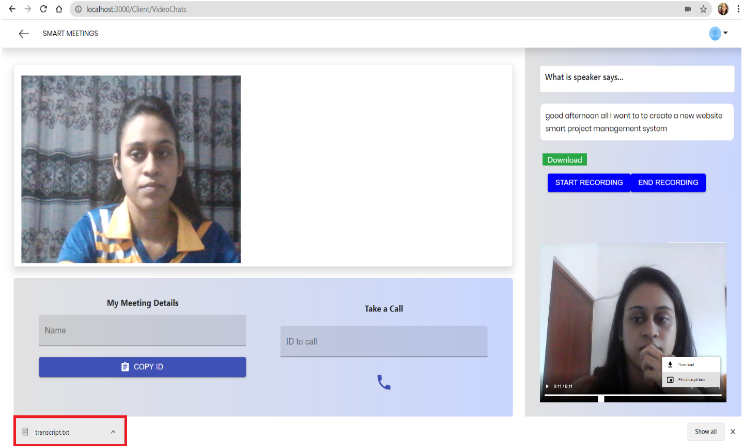
Above Table represents the GitHub contribution prediction generated using the proposed system. Since the system considers the data from the previous weeks, next prediction should be according to the previous data and using these predictions, supervisors and the student himself can identify if project completion is possible. However, since there are no researches conducted by researchers for GitHub contribution prediction, this is the first research that GitHub contribution added to a project management system.

When we consider about the display all projects to a relevant logged in user, all the projects are display for each logged in user (Figure 13 in the implementation section). For the single project display project data page also displaying data successfully (Figure 14 in the implementation section).

### **3.1.3 Client Meeting tracker**

In this system mainly focus the Client Meeting Platform and Content Documentation of the overall meeting and satisfaction rate of client. First of all, we create a meeting platform that will allow the students to carry out video meet-up with the client to analyze the project. After that The system will do a real-time voice-to-text converting and shows them up on the screen. Also, the system can record the meeting in audio format, which can be retrieved after the meeting.

As a result, the system will produce the report of the meeting which consists of text conversions done by the system and calculate the client's satisfactory rate using emotional detection from the text or emotion. Finally, the system generates reports and allow to download those reports and video for future use. Figure 52 show the recorded video of the meeting.



##### Figure 52: recorded video of the meeting

Graphical user interface, application, Word

Description automatically generated

##### Figure 53: Downloaded transcript

### **3.1.4 Peer Review**

In this result section we are going through results related to all the models in the algorithm in this first segment we can see the topic generation f the LDA base model to given topic count, in this base model alpha values are in auto status.

Table 22: LDA base model parameters

|  |  |
| --- | --- |
| corpus=corpus, | chunksize=100, |
| id2word=id2word, | passes=10, |
| num\_topics=num\_topics, | alpha='auto', |
| random\_state=100, | per\_word\_topics=True |
| update\_every=1, |  |

[(0,

'0.024\*"much" + 0.024\*"leader" + 0.024\*"team" + 0.024\*"time" + '

'0.024\*"choice" + 0.024\*"talk" + 0.024\*"perfect" + 0.024\*"good" + '

'0.024\*"right" + 0.024\*"well"'),

(1,

'0.159\*"leader" + 0.159\*"much" + 0.083\*"role" + 0.083\*"perfect" + '

'0.083\*"pretty" + 0.083\*"active" + 0.083\*"good" + 0.008\*"team" + '

'0.008\*"choice" + 0.008\*"talk"'),

(2,

'0.090\*"weakness" + 0.090\*"aleader" + 0.090\*"perfect" + 0.090\*"see" + '

'0.090\*"doubt" + 0.090\*"choice" + 0.090\*"team" + 0.090\*"leader" + '

'0.008\*"much" + 0.008\*"talk"'),

(3,

'0.025\*"time" + 0.025\*"team" + 0.024\*"conversation" + 0.024\*"bit" + '

'0.024\*"hard" + 0.024\*"group" + 0.024\*"deadline" + 0.024\*"make" + '

'0.024\*"cause" + 0.024\*"decision"'),

(4,

'0.123\*"much" + 0.083\*"team" + 0.083\*"leader" + 0.083\*"talk" + 0.044\*"time" '

'+ 0.044\*"choice" + 0.044\*"good" + 0.044\*"group" + 0.044\*"know" + '

'0.044\*"person"'),

(5,

'0.024\*"leader" + 0.024\*"much" + 0.024\*"team" + 0.024\*"talk" + 0.024\*"time" '

'+ 0.024\*"perfect" + 0.024\*"right" + 0.024\*"choice" + 0.024\*"project" + '

'0.024\*"group"'),

(6,

'0.104\*"team" + 0.104\*"time" + 0.054\*"active" + 0.054\*"much" + '

'0.054\*"little" + 0.054\*"remind" + 0.054\*"decision" + 0.054\*"cause" + '

'0.054\*"make" + 0.054\*"deadline"'),

(7,

'0.109\*"quickly" + 0.057\*"hope" + 0.057\*"function" + 0.057\*"difficulte" + '

'0.057\*"conversation" + 0.057\*"contact" + 0.057\*"bit" + 0.057\*"difficutie" + '

'0.057\*"little" + 0.057\*"figure"'),

(8,

'0.024\*"leader" + 0.024\*"much" + 0.024\*"team" + 0.024\*"talk" + 0.024\*"good" '

'+ 0.024\*"choice" + 0.024\*"perfect" + 0.024\*"time" + 0.024\*"right" + '

'0.024\*"well"'),

(9,

'0.024\*"team" + 0.024\*"leader" + 0.024\*"much" + 0.024\*"talk" + '

'0.024\*"choice" + 0.024\*"time" + 0.024\*"perfect" + 0.024\*"good" + '

'0.024\*"little" + 0.024\*"right"')]

We are also calculating each model training duration.in this base model its training duration is:

--- 0.04599738121032715 seconds ---

After calculating the training duration, we are calculating the coherence scores with cv and UMass metrics.

Graphical user interface, application, Word

Description automatically generated

##### Figure 54: Cv score calculation of the LDA base model

**Graphical user interface, text, application

Description automatically generated**

##### Figure 55: Umass calculation of the base model

As mention in the methodology, lower perplexity scores demonstrate the best algorithms in language models. After training time and coherence score calculations algorithm moves on to the data visualization stage with the PyLDAvis library. We will go through each topic counts intertopic distance maps with multidimensional scaling and the relevant token frequency maps with saliency and relevance metrics.

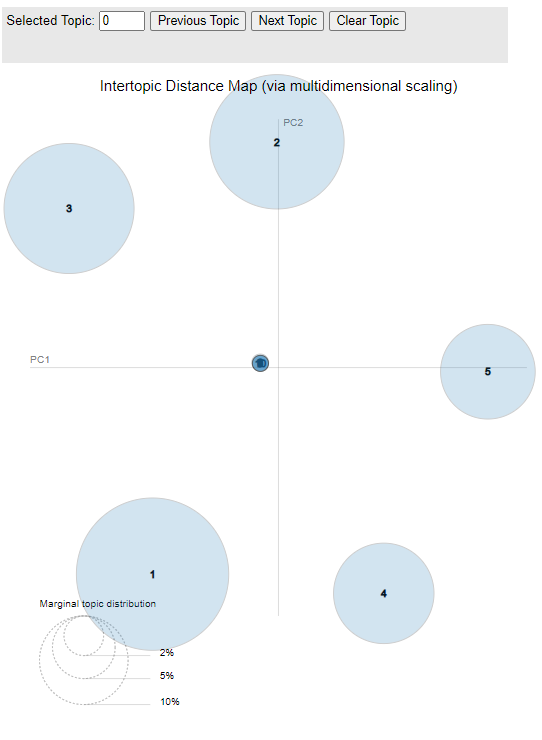
Saliency is defined as below. First We need computational probability. For a word w its computational probability which is P(T|w), the likelihood of the word w is generated by T latent topic. We also need marginal probability. Which is the likelihood of random-word(w`) generated by T latent topic.[34]

Distinctiveness is kullback-leibler divergence[35].to calculate the saliency we need distinctiveness measurements with computational and marginal probability values.

distinctiveness(w) = X T P(T|w) log P(T|w)/ P(T)

in information theories, we can describe it as generated topics information contribution to the document which it exists with its frequency. Saliency is needed to generate the term-topic matrix which we can see in the below figure.

saliency(w) = P(w) × distinctiveness(w)

****In this figure 33, we have the initial intertopic map with none of the topic selected. Its

##### Figure 56: Initial intertopic distance map

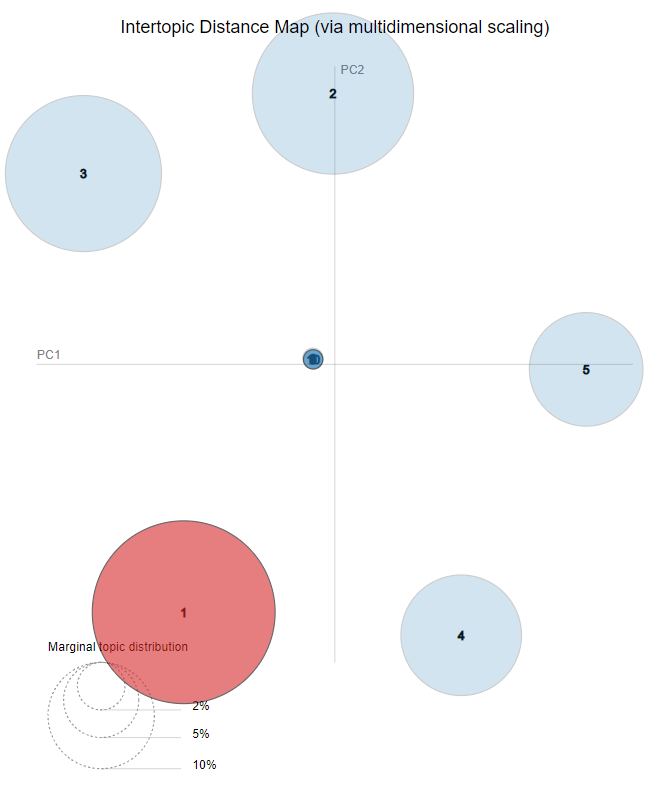
demonstrate the intertopic dimensions before topics go through the map. As we can see estimated term frequency does not exist in the initial map.

**Graphical user interface, application

Description automatically generated**

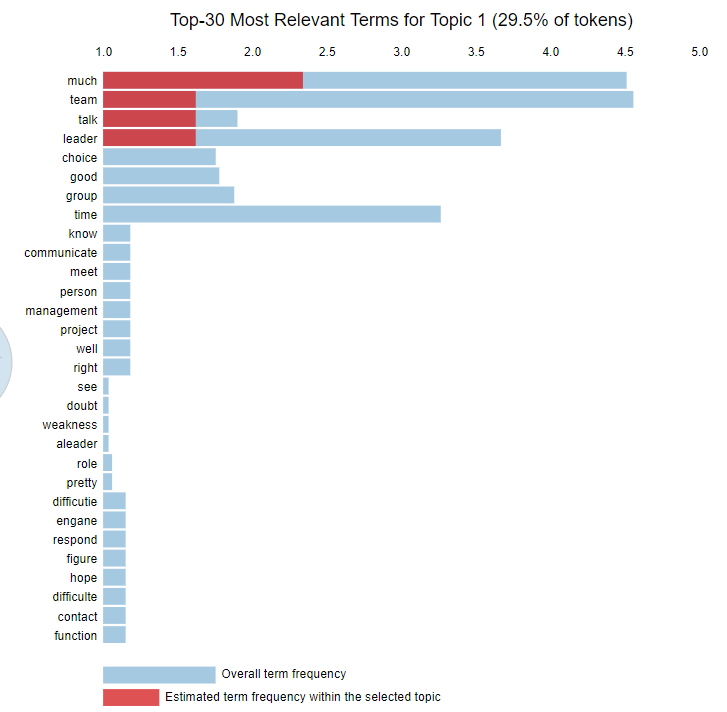
##### Figure 57: Term-topic distribution

In this figure we can also see the estimated term frequency is not existing like on the map. Also term related to overall frequency only exist with its defined relevance score.

****

##### Figure 58: Intertopic distance map first iteration

In this figure 58, we can see the first iteration of the term-topic distribution. in this map, we have estimated term frequency indicates in the first word topic cluster. Like that term frequency will goes through each cluster according to their relevance and saliency scores.

****

##### Figure 59: Term-topic distribution first iteration

In this figure 59, this term frequency allocation it indicates the first topic estimated term frequency with the overall term frequency illustrations. This is a direct adaption from the map which both use pyLDAvis library.

**Text, letter

Description automatically generated**

##### Figure 60: gensim LDA mallet topic generation pattern

Figure 60 demonstrates the gensim LDA mallet topic generation pattern up to selected 3 topic variations it can set to prefer count .in each topic cluster most probability calculated word will be represented as the topic according to its term-topic frequencies.

Its training time calculation is demonstrated below figure as same as the base model training time. It spends 3 more seconds than the base model which is not big different when it comes to individual review processes.

--- 48.57359981536865 seconds ---

If we consider the overall result, Each and every component executed as expected and expected results were returned from the system. After the unit testing session, We’ve continued with the integrated system testing after the integration phase. We’ve executed the same tests with the integrated system and it performed as we expected.

## **3.2 Research Findings**

### **3.2.1 Group Generation**

Primary goal of this research is to introduce a group generation environment to the project management systems. Most of project management systems are developed for common requirements, such as task management, maintain workflow and much more advanced procedures. Since most of existing project management systems doesn’t include this facility, it is valuable to have a group generation system in a project management system to a university environment.

During our initial research, we’ve found that most of universities currently using manual ways of group generation. Even from our experience, it is time consuming and exausting process for both management and students. So the goal was to develop a group generation functionality and merge the functionality with a newly created project management system where universities can manage the system by themselves.

After indentifying the problem, we’ve proposed a solution to develop a group generation algorithm. But in order to develop such algorithm, we had to decide which attributes and data we consider. So we did a requirement gathering and analysis. From what we’ve learned, we can easily develop and grouping algorithm using GPA of students. Since GPA is a main factor that universities consider to measure a student’s skill levels, we’ve decided to use the GPA as a contributing factor to the grouping algorithm. But using only GPA we can not decide student’s capability. So we’ve decided to extract student skill details from them.

In order to extract student’s skills, we’ve used a google survery. Using the survey, we were able to extact student skill data. This included various types of skills. Such as development technology, development media and programming languages. Then we figured, these attributes are more than enough to start development of the algorithm.

Compare to previous researches, main advantage of the system is, it is integrated with the project management system. By doing so, we can easily manage student progress and group progress by using the same system. Since the grouping algorithm combined with stabalizers, it is much more reliable, effective and efficient. Table 1 represents the result comparison between previous researches and other strategies.

Table 23: Comparison between previous systems

| Grouping Approach | Grouping Strength(Average GPA) | Group Count | Students per group |
| --- | --- | --- | --- |
| By Hand | 3.02 – 3.31 | 9 | 4 \* 5 + 3 \* 2 + 2 \* 2 |
| Proposed | 3.23 – 3.53 | 8 | 4 \* 6 + 3 \* 2 |
| Team-Maker | 2.9 – 3.46 | 8 | 4 \* 7 + 2 \* 1 |
| GroupEng | 3.11 – 3.27 | - | - |

The final solution is much more reliable to use and it is much more effective and efficient on group generation.

### **3.2.2 GitHub prediction and progress tracking**

The key goal of this research was to predict future contribution per each student and progress tracking.

While researching this topic, we identified, the accuracy of this research, predict future contribution, can’t use in this research. Because this is just a prediction and the prediction process work with the last weeks as a prediction for the next week. So we display actual commits and the prediction commits for each student. We identified that was the way to predict each student contribution using commits contribution.

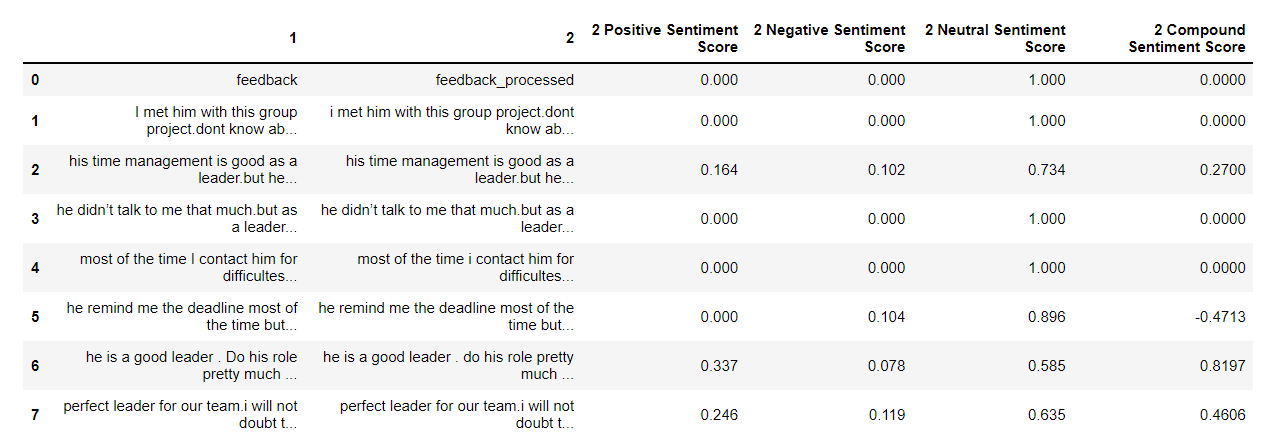
When we consider about progress tracking, we identified using GRA, can track project and the contribution. Only student has to pass the PAT, Repository name, Owner name and the GitHub URL. Using that data passing, system can track the project.

If the findings are compared with the existing research. Here, the algorithms used are very effective. While this analysis covered the newest developments in the application growth. Because this research is a new finding. The application is very good as it responds with less than a few seconds. This proposed system can also be trusted by consumers.

### **3.2.3 Peer Review**

After optimal model build and training algorithm generating the dominant topic coverage with its relevant keywords. This table has document numbers provided, dominant topic unit, topic contribution, topic percentage contribution, and the keywords for each contributed topic.

##### Diagram Description automatically generated with low confidenceFigure 61: Topic number coverage

****After the dominant topic coverage algorithm demonstrates the topic numbers with topic percentage contribution according to their keyword for each topic contribution.

##### Figure 62: Sentiment score coverage

After the main algorithm end with its coverage reports we algorithms helper function will generate the sentiment scores according to pre-processed data from the early stages to compare the sentiment scores with given student’s review scores. Below we listed the score variation that can be occur with the algorithm functionality.

Table 24: Basic review flow

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Review score | Positive sentiment score | Neutral sentiment score | Negative sentiment score | Compound sentiment score |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

### **3.2.4 Client Meeting tracker**

The main goal of this research is to introduce a speech recognition and calculate the client satisfaction rate of emotional detection. Most of the existing system of Speech Recognition and Emotional Recognition are used and get the outputting only a result that is related to face or speech. However, in our research, we will create that by combining those two recognition systems into a single system, calculating the customer satisfaction rate, and ultimately generating the report. So we found the Emotional and Speech recognition systems currently running the system’s algorithms, Research papers related to my part, what are existing Recognition Toolkit are used, find out Basic rules and methods of Speech recognition technologies.

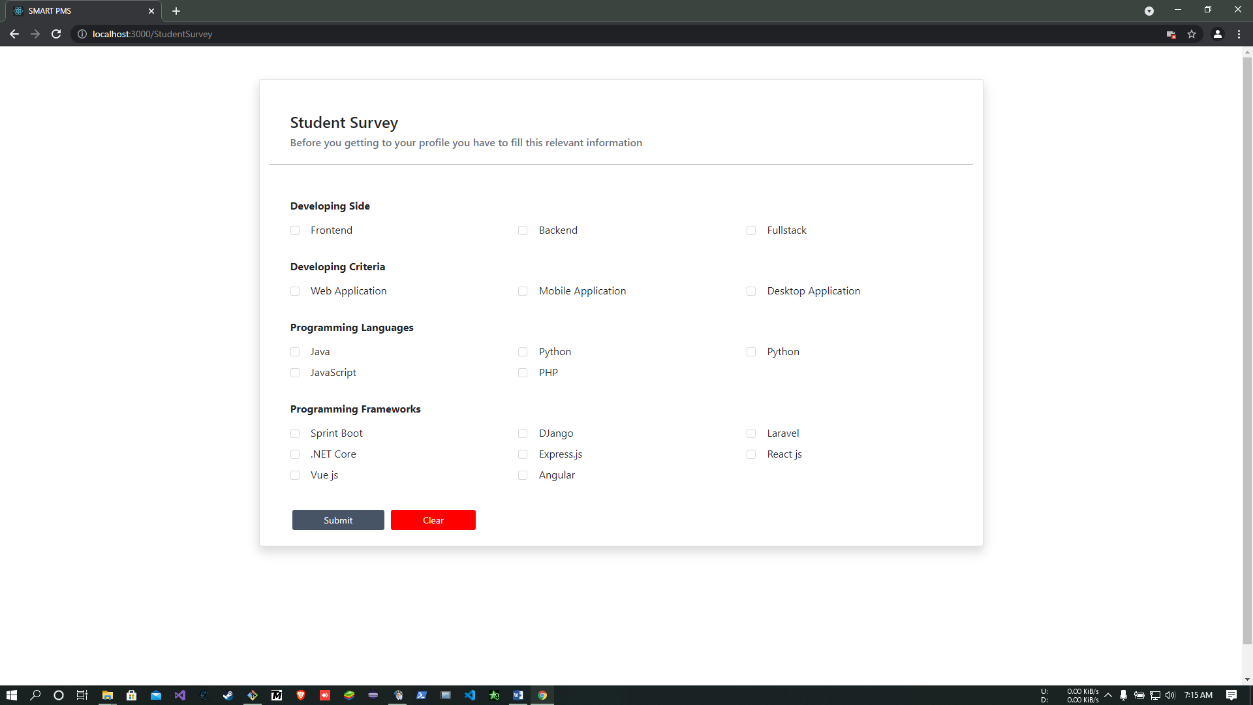
After identifying the shortcomings of existing systems, we presented a method to integrate these two functions into a single system. As a result, it is quite beneficial for both clients and students to manage their requirements. since they can finally obtain a sense of the client satisfaction rating and provide information. So we do our implantations using Speech Recognition and WebkitSpeechRecognition library modules in JavaScript, and for the emotion recognition on the client feedback part, we have tried using the face-API. JS library from JavaScript and implement our research. Compare to previous researches, the main advantage of the system is we can get final output and the system can record the meeting in audio format, which can be retrieved after the meeting and at last, the system will produce the report of the meeting which consists of text conversions done by the system and calculate the client's satisfactory rate using emotional detection from the text or emotion. So those all functions are working in one system.

Final results from each component shown effective and efficient work flow. From group generation, we were able to generate student groups way more efficient that existing solutions. By introducing the GitHub contribution predictor, we were able to identify student’s dedication to the project as well as their contribution. Client portal was a huge for students, since it document a summary of the complete meeting session content. Peer review section helps users to identify free riders and also from the reports that generate from the system can be used in evaluation phases.

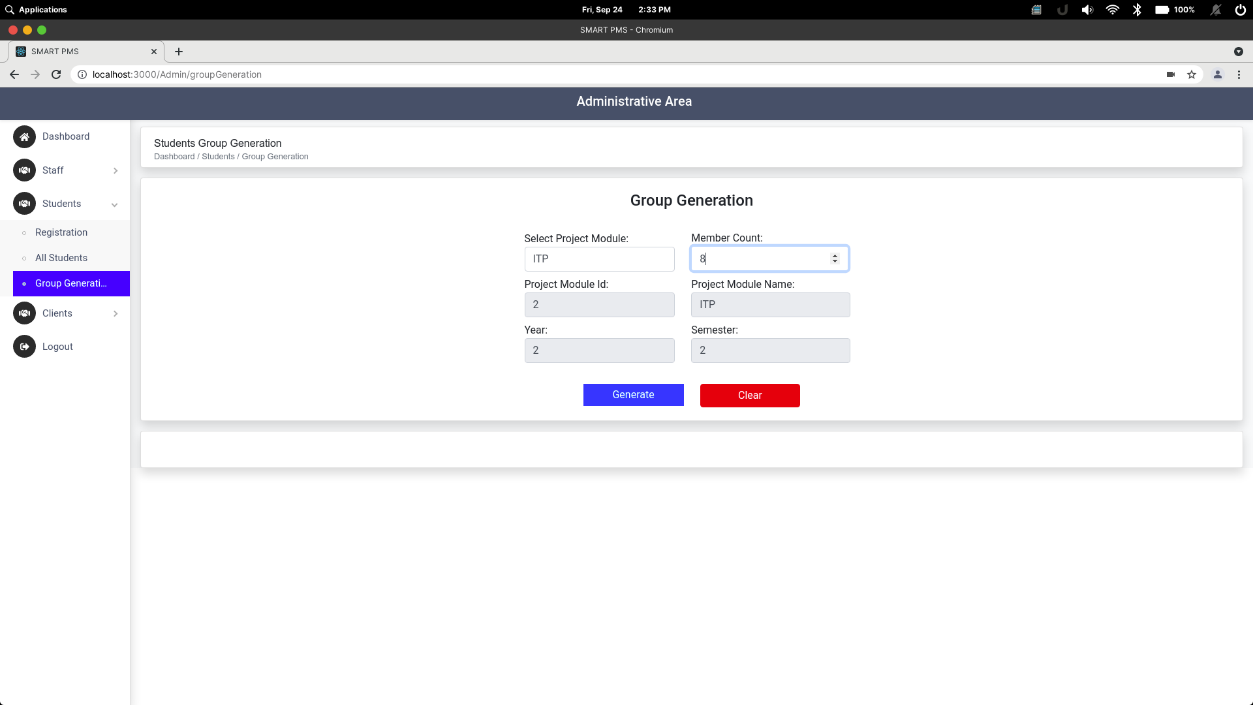
Each component contributed to the project management workflow. From Registration to the evaluation, system generates reports and other necessary details for the supervisors.

## **3.3 User Interfaces**

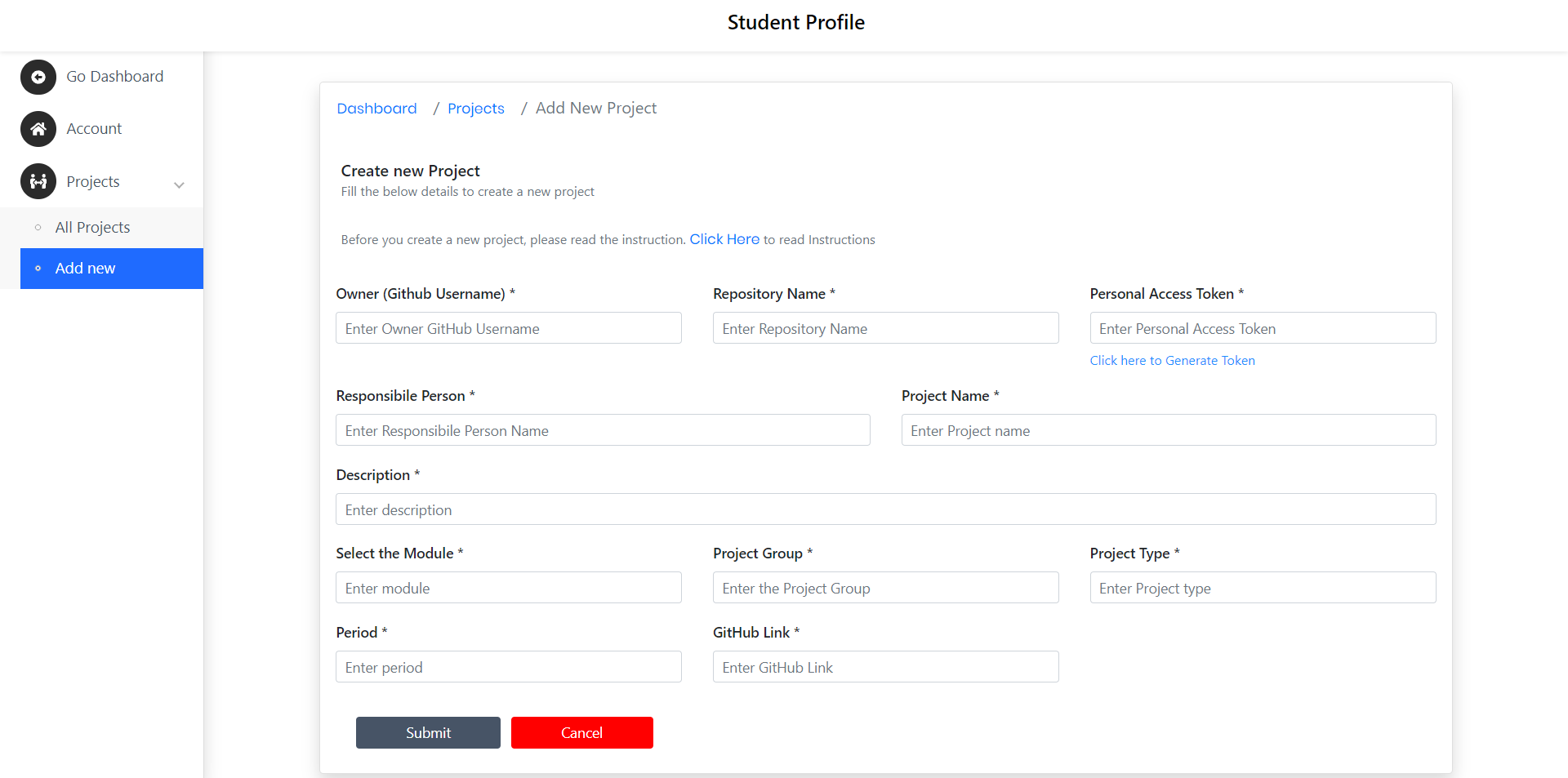
This chapter introduce the User Interfaces that relevant to Group Formation Process.



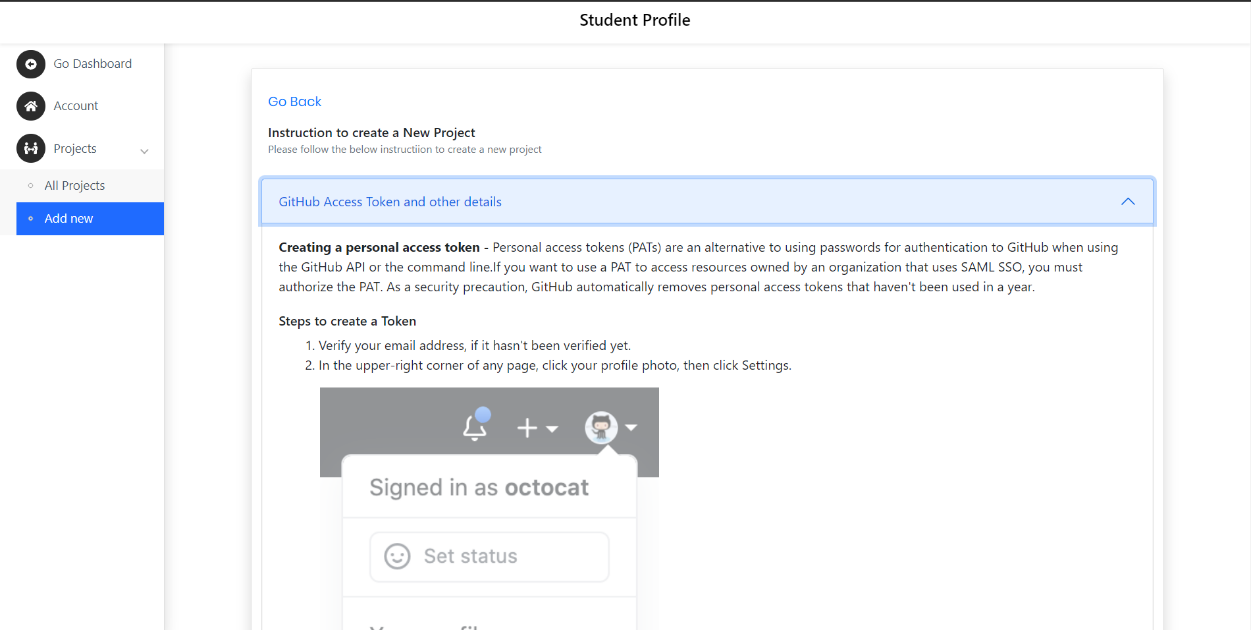
##### Figure 63: Student Survey



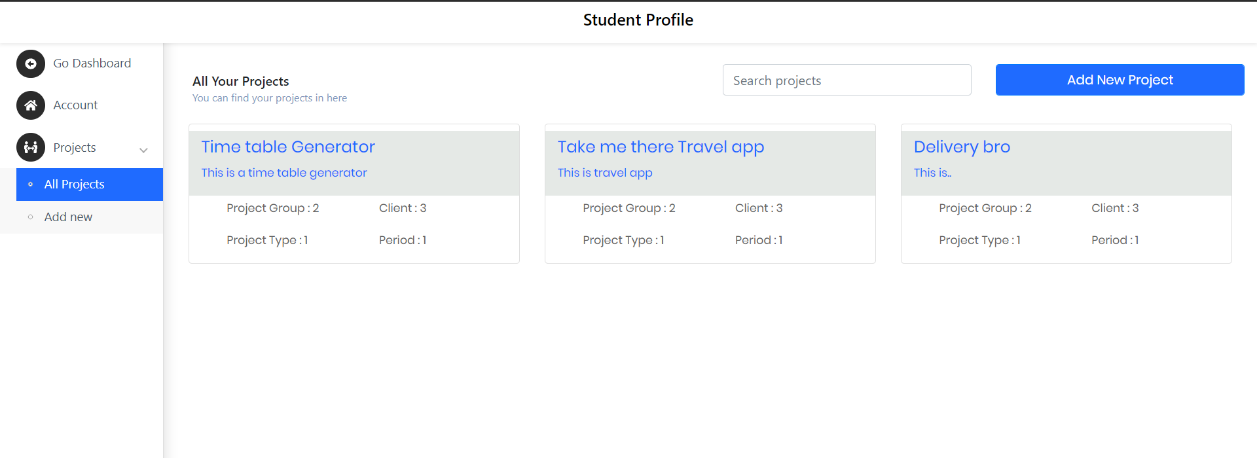
##### Figure 64: Student Group Generation UI



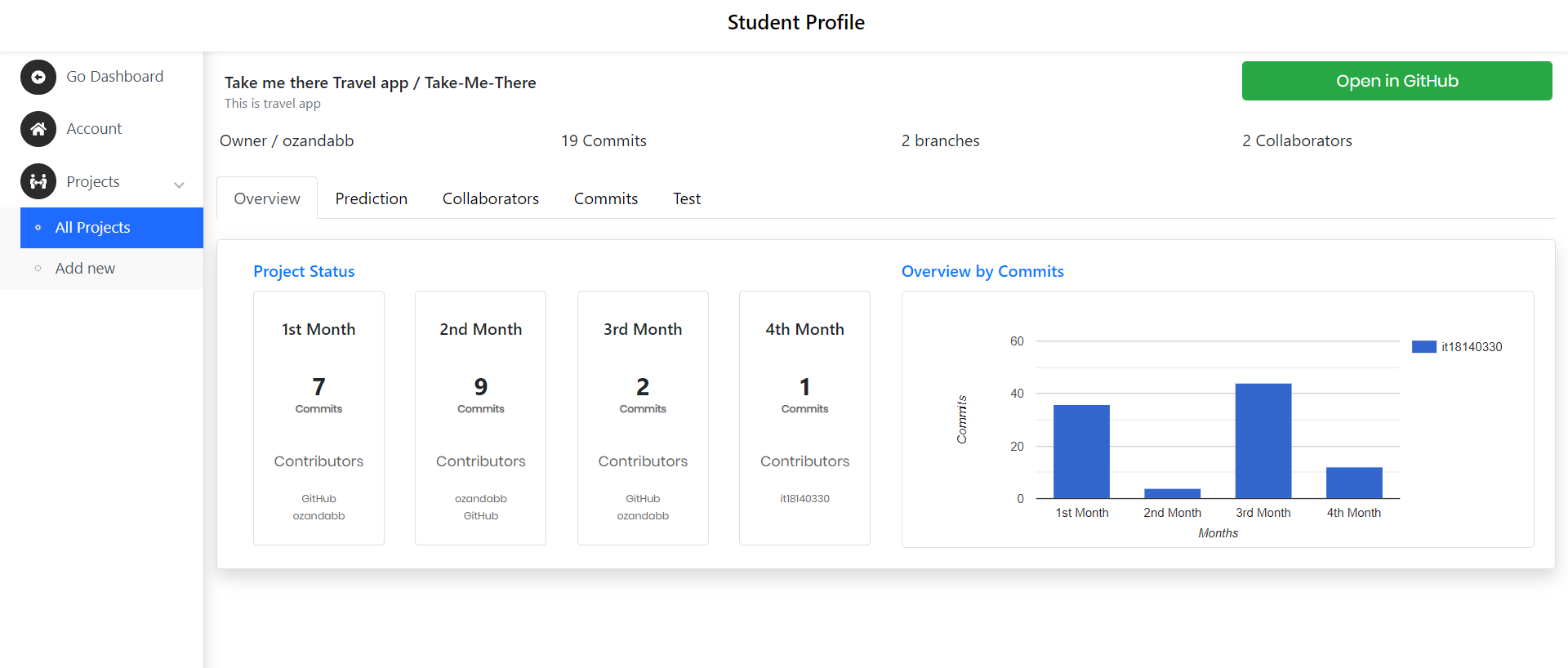
##### Figure 65: create new project environment



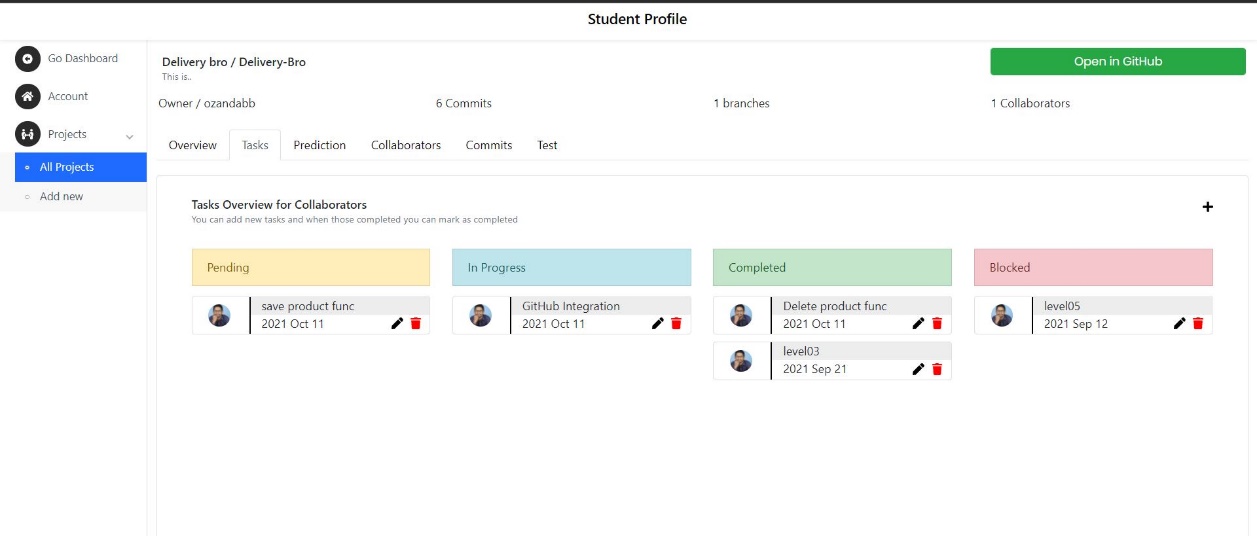
##### Figure 66: screen for how to create Personal Access Token



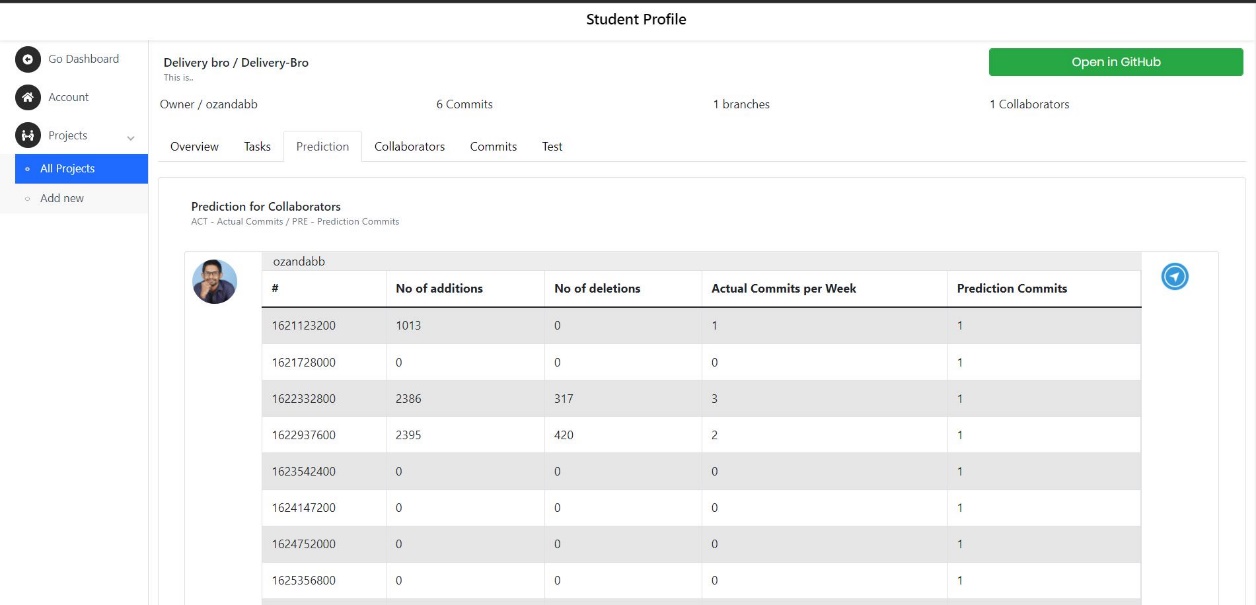
##### Figure 67: all projects



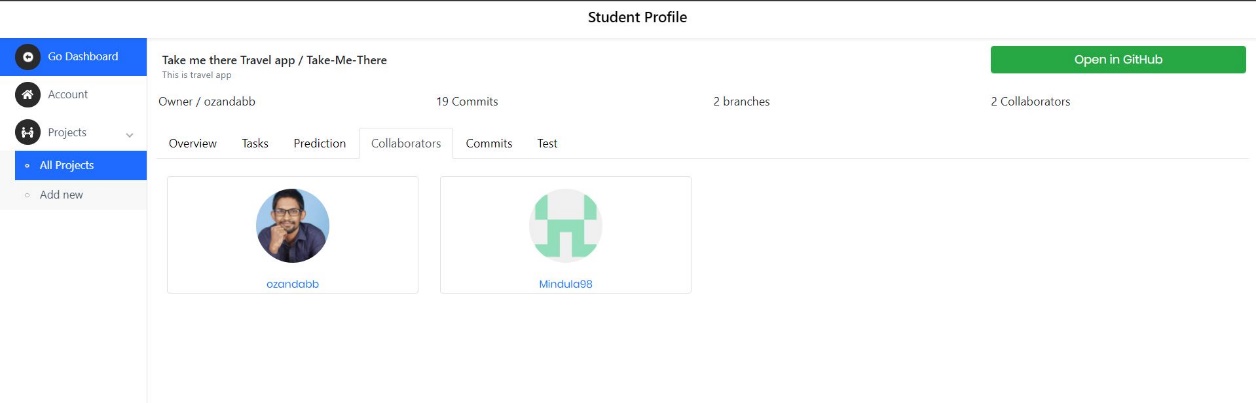
##### Figure 68: single project view



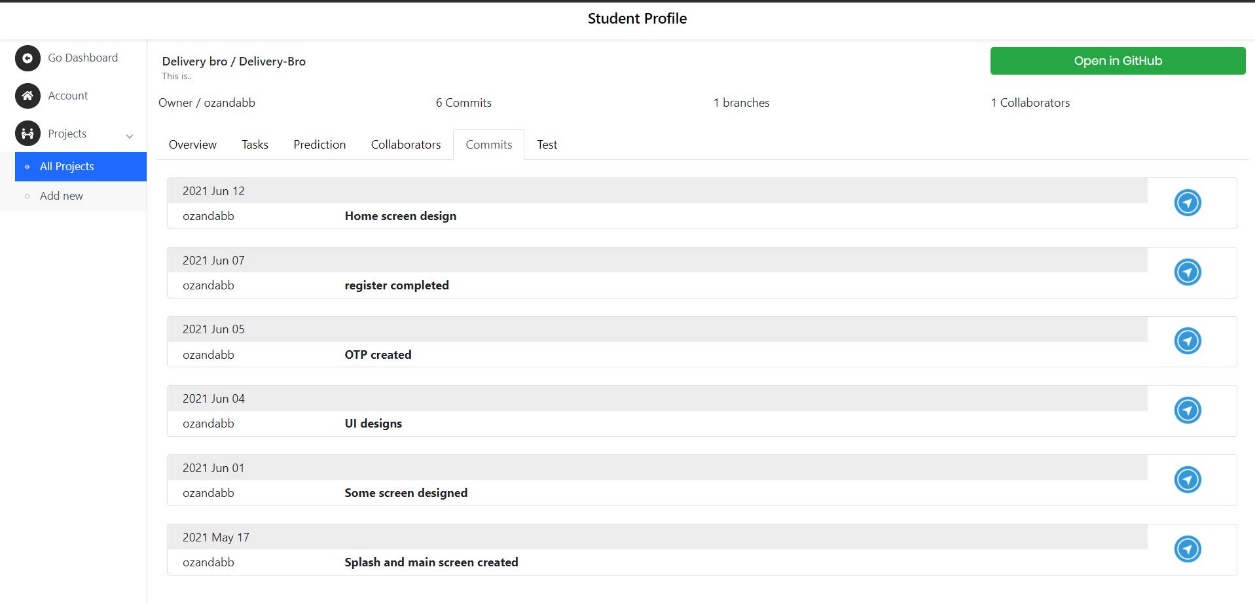
##### Figure 69: task dashboard



##### Figure 70: prediction tab for each member



##### Figure 71: all collaborators



##### Figure 72: all commits

## **3.4 Discussion**

As a smart project management system its methodology, implementation and results prove its stand with specific, measurable, attainable, realistic and timely boundaries to fulfill its smart functionality. its specific, it has its own target audience which is student who are doing their undergraduate studies and higher education studies. Its measurable, group generation, group progress, client satisfaction, peer review marks all these component represent measurements. Its attainable which we demonstrate and achieved proposed components functionality within given time period. It’s realistic, we backed our research study with the background and literature and build the logic as novice as we can. its timely, again with components process oriented functionality its specifically timely solution on its own.

We proposed these components to improve the existing current project management systems. Some of the project management systems not even have these components such as group generation. With group generation automation strangers might get team up.so the progress and friendly environment won’t be same as teaming with our friends which is the current existing methods to form a group if it not forms group based on some order. In this research studies, we prove both those strategies are performing below automatic group generation.

As mentioned above automatic group generation leads to empowering the progress monitoring because of the freshness of the teammates.in progress tracking we are empowering the version control directly with the beginning of the project module. There are lots of workshops, lectures, tutorials, labs to empower the usage of version controlling in the project among students but this would be the most effective way of those all and in this research studies we prove it. It’s get insight to themselves is the best outcome of it because it is acting motivator to the project and its members. Students can be not go often to version control platform but their coming to the project management system more often within the project timeline and its empowering the version controlling with these progress tracking which will lead them to head to version control system because of the forcing motivation it gives to them with information gathering.

Client satisfaction rate and meeting platform is the practice of bringing individual existing project management tools and merge to our project management system. Emotion recognition with it is the novice approach to gather client’s insight about student’s engagement with him or her which will lead again to empower the process oriented project management practices.

Peer-review assessments are heavily use on industry and the web platform to improve the products and processes. it’s another novice approach to detect free-riding effect variations on the project management process which is currently have no active answers on the project management in the undergraduate modules. This component empower the outcome based education on process oriented approach.

# 4. SUMMARY OF THIS RESEACH CONTRIBUTION

Table 25: Description of personnel and facilities.

|  |  |  |
| --- | --- | --- |
| **Member** | **Component** | **Task** |
| IT18009446  Amarasekara T.N.E. | Student group generation in order to make student group fairly and efficiently. | * Collect data to initiate algorithm development. * Create a solution to collect student skills and data interactively. * Use machine learning approach to identify connection between data types. * Create an algorithm to generate student groups. * Create a solution to view outputs and give an overview of the overall groups. |
| IT18129236  Isurindi H.G.P. | Tracking using voice-to-text converter and Emotional recognition to calculate client satisfaction rate | * Develop voice-to-text converter which generates a report. * Develop client satisfaction rate calculator using Emotional recognition as a text * Develop client meetings platform |
| IT18140330  Gamage O.M. | Project Progress tracking and predict future contribution. | * Assign task or a topic by a lecturer or supervisor part * Track the GitHub project progress Using GitHub REST API * Generate a report related to project progress * Predict future contribution using supervised learning algorithm * Issue tracking Send Notification about incoming events and due dates |
| IT18003642  Navanjana E.H.D.T.D. | Peer-review assessments and detecting free-riding effect variations | * Develop component to collect reviews from the students * Build and train main algorithm to visualized topic distribution * Build the helper algorithm to calculate the sentiment score * Build and train main algorithm using hyper parameter tuning * Generate report using main algorithm and helper algorithm outcomes to visualize the results |

# 5. CONCLUSION

With the proposed project management system was able to manage the workflow of a common project in a university environment. Combined with major requirements for a university projects, system was able successfully execute a complete project management process workflow. Starting with group generation, GitHub contribution predictor predicts future contributions from the students. Furthermore, with the meeting documentation process, students can check previous meeting details. And finally the peer review was able to contribute for the evaluation phase of a project management process. Since the proposed project management system was able to cover major steps of project management.

Even so the system is functional as proposed, system can be further improving with new technologies and features. Most of project management systems that currently in use are mostly for common purposes. Developing a configurable project management system is much more valuable for institutes since they have different variations of requirements. By developing such system is much more effective for an educational environment.

Starting from the student group generation, students can learn from other students since the grouping process not only creating well balanced groups, but also create diverse teams. This grouping process can be improved by increasing the number of variables that used to decide which student will be most valuable to a selected group. Increasing the number of variables will increase the diversity among groups which helps to create well balanced groups. By using the client portal, system can be updated to help students to improve the productivity by designing solutions for external clients. Which will be a direct effect on their educational skills and also for their industrial experience. Likewise, the GitHub contribution predictor and peer review features can be improved where it can be used as a professional certificate. Which can be used in an interview, where student can present their contribution for projects inside the institution and by presenting their reviews that given by their supervisors. Which leads to the main conclusion, which is the proposed system is much more valuable than regular project management systems where it not only helps to manage project but also students can use their overall performance as a certified result from their institution.

# 6. REFERENCES

[1] Chen Di and Jiang Bei-zhan, “Some Problems in Teaching ‘Educational Technology in Project Teaching’”, Journal of e-Education Research,pp.63-65 2007.

[2] Dimiduk, T. G., & Dimiduk, K. C. (2011, December). Effectively assign student groups by applying multiple user-prioritized academic and demographic factors using a new open source program, GroupEng. In 2011 WEPAN Nat. Conf. Advancing Women: Transforming Eng. Educ (pp. 1-12).

[3] Chirag Mehta, & Tamara Swedberg, Team Maker, 2000. [online]. Available: <https://chir.ag/projects/team-maker/>. [Accessed: 19-Feb- 2021]

[4] Brant Gurganus, Team Maker, 2000. [online]. Available: <https://sourceforge.net/projects/team-maker/>. [Accessed: 19-Feb- 2021]

[5] Z. Wen-qiang and W. Ben-ting, "Project practical teaching grouping strategy based on fuzzy clustering algorithm," 2010 5th International Conference on Computer Science & Education, Hefei, China, 2010, pp. 678-681, doi: 10.1109/ICCSE.2010.5593662.

[6] Livermore, J. A. (2007, March). Factors that impact implementing an agile software development methodology. In Proceedings 2007 IEEE SoutheastCon (pp. 82-86). IEEE.

[7] ReactJS. (2021). A JavaScript library for building user interfaces. [online]. Available: <https://reactjs.org/> [Accessed 21-Feb-2021]

[8] Paul Krill, “React: Making faster, smoother UIs for data-driven Web apps”, May 15, 2014. [online]. Available: <https://www.infoworld.com/article/2608181/react--making-faster--smoother-uis-for-data-driven-web-apps.html>. [Accessed: 20-Feb-2021]

[9] Hemel, Zef, "Facebook's React JavaScript User Interfaces Library Receives Mixed Reviews", Jun 03, 2013. [online]. Available: <https://www.infoq.com/news/2013/06/facebook-react/> .[Accessed: 20-Feb-2021]

[10] Dawson, Chris, “JavaScript’s History and How it Led To ReactJS”, July 25, 2014. [online]. Available: <https://thenewstack.io/javascripts-history-and-how-it-led-to-reactjs/> .[Accessed: 20-Feb-2021]

[11] Quarkus. (2021). A Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards. [online]. Available: <https://quarkus.io/>. [Accessed: 21-Feb-2021]

[12] The Apache Software Foundation. (2021). What is maven. [Online]. Available: <https://maven.apache.org/what-is-maven.html>. [Accessed: 21-Feb-2021]

[13] Javatpoint. (2018). JDBC Driver. [online]. Available: <https://www.javatpoint.com/jdbc-driver>. [Accessed: 21-Feb-2021]

[14] Postman. (2021). The Collaboration Platform for API Development. [online]. Available: <https://www.postman.com/>. [Accessed: 21-Feb-2021]

[15] Amazon Web Services. (2021). Cloud computing with AWS. [online]. Available: <https://aws.amazon.com/what-is-aws/?nc1=f_cc>. [Accessed: 21-Feb-2021]

[16] Srba, I., & Bielikova, M. (2014). Dynamic group formation as an approach to collaborative learning support. IEEE transactions on learning technologies, 8(2), 173-186.

[17] Loughborough University, Web-PA, 2006. [Online]. Available: <https://webpa.lboro.ac.uk>. [Accessed: 23- Jan- 2021]

[18] Project Management Institute 2008. A Guide to the Project Management Body of Knowledge (PMBOK Guide) (4rd ed.). Project Management Institute.

[19] Gousios, G., Vasilescu, B., Serebrenik, A., & Zaidman, A. (2014, May). Lean GHTorrent: GitHub data on demand. In Proceedings of the 11th working conference on mining software repositories (pp. 384-387).

[20] GitHub Documentation Guide, ”REST API”, Available at <https://docs.github.com/en/rest>

[21] GitHub Documentation Guide, “Creating a personal access token” , Available at <https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token>

[22] Chow, Y., Dunham, M., Kimball, O., Krasner, M., Kubala, G., Makhoul, J., ... & Schwartz, R. (1987, April). BYBLOS: The BBN continuous speech recognition system. In ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 12, pp. 89-92). IEEE.

[23] Borges, H., Hora, A., & Valente, M. T. (2016, September). Predicting the popularity of GitHub repositories. In Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering (pp. 1-10).

[24] Kikas, R., Dumas, M., & Pfahl, D. (2016, May). Using dynamic and contextual features to predict issue lifetime in GitHub projects. In 2016 ieee/acm 13th working conference on mining software repositories (msr) (pp. 291-302). IEEE.

[25] Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Millier, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017.

[26] IEEE, “A facial expression emotion recognition based human-robot interaction system” , 2017

[27] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes, and Blenders: Domain Adaptation for Sentiment Classification. In ACL.

[28] Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In ACL.

[29] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based LSTM Network for Cross-Lingual Sentiment Classification. In EMNLP.

[30] Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large­Scale Sentiment Analysis for News and Biogs. In International Conference on Weblogs and Social Media.

[31] Exploring popular topic models, Poonam Tijare and P Jhansi Rani 2020 J. Phys.: Conf. Ser. 1706 012171

[32] S. Syed and M. Spruit, "Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation," 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2017, pp. 165-174, doi: 10.1109/DSAA.2017.61.

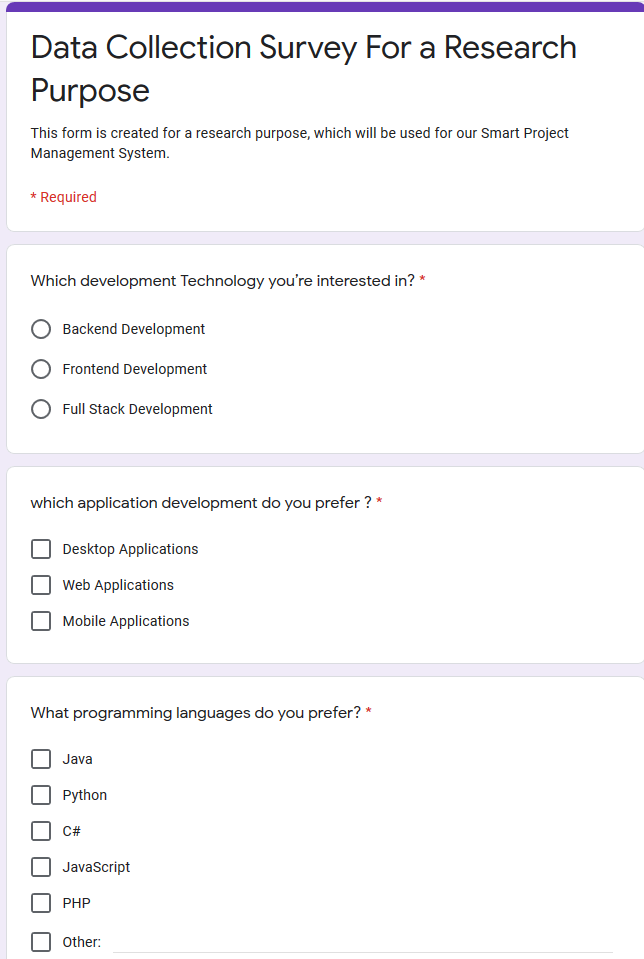
[33] Röder, Michael et al. “Exploring the Space of Topic Coherence Measures.” Proceedings of the Eighth ACM International Conference on Web Search and Data Mining (2015): n. pag.

[34] Termite: Visualization Techniques for Assessing Textual Topic ModelsJason Chuang, Christopher D. Manning, Jeffrey Heer Stanford University Computer Science Department

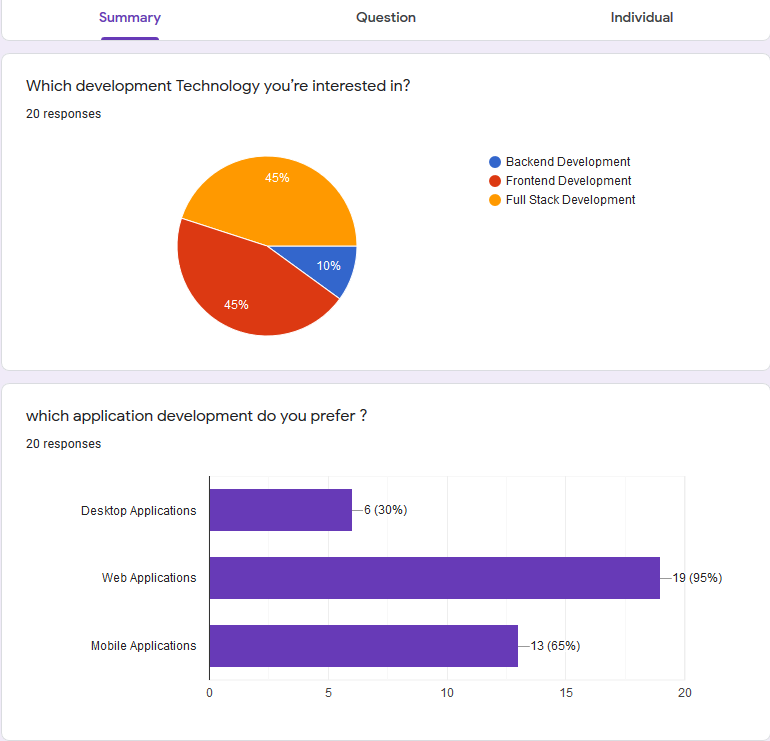
[35] S. Kullback and R. A. Leibler. On information and sufficiency.Annals of Mathematical Statistics, 22(1):79–86

# 7. APPENDICES

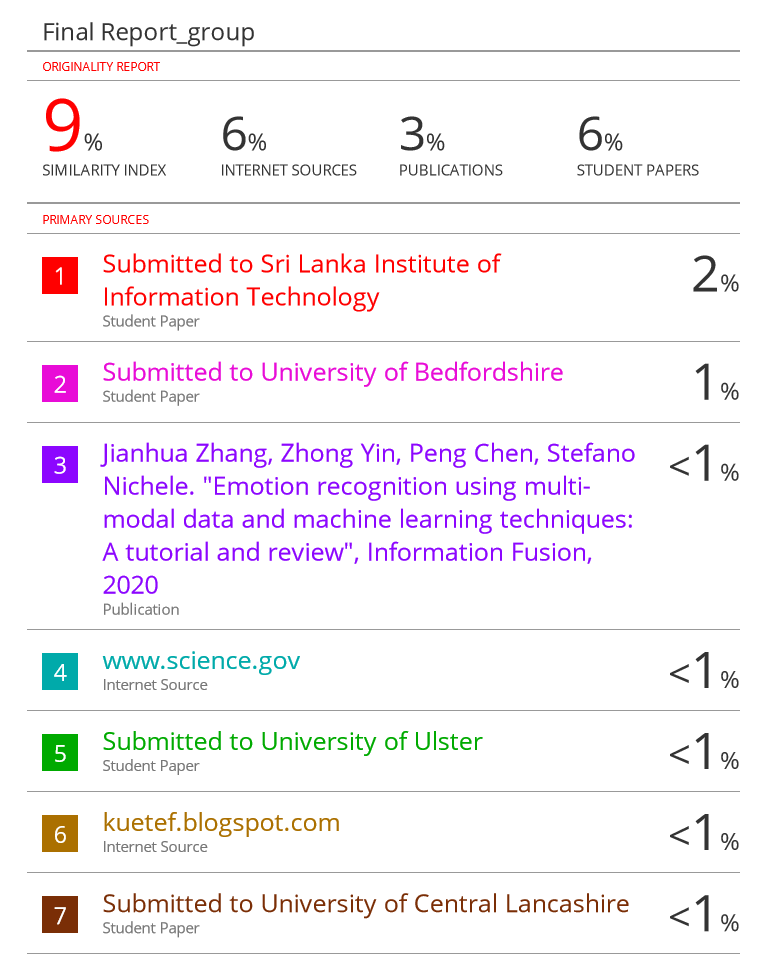
## Appendix – A: Sample questionnaire



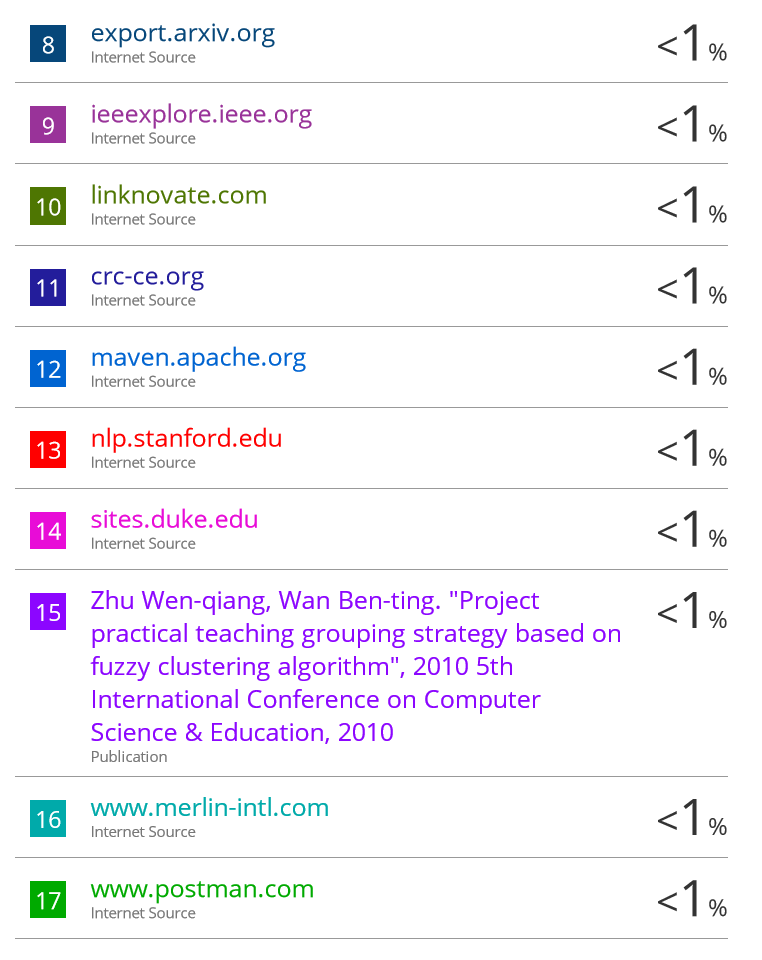
## Appendix – B: Sample questionnaire response



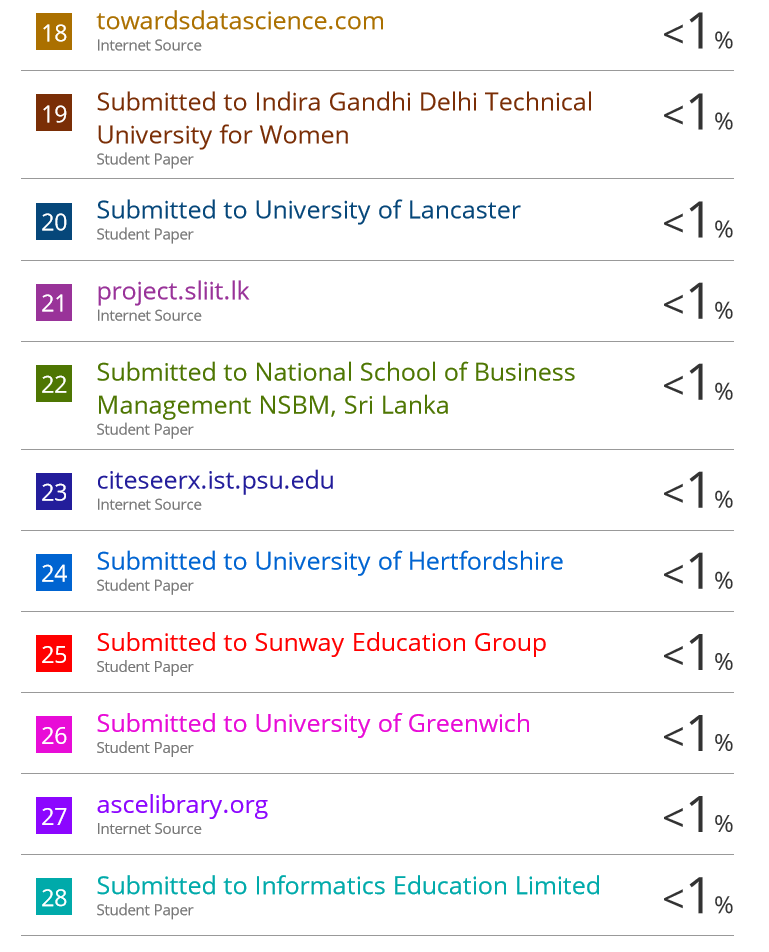
## Appendix - C: Plagiarism Score 01



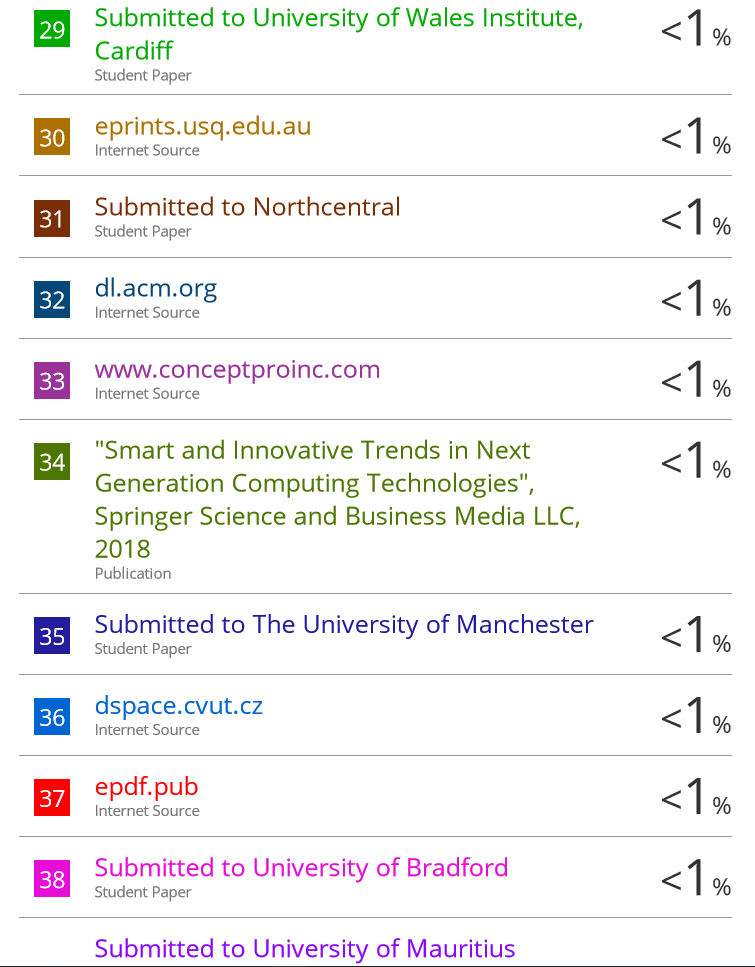
## Appendix - D: Plagiarism Score 02



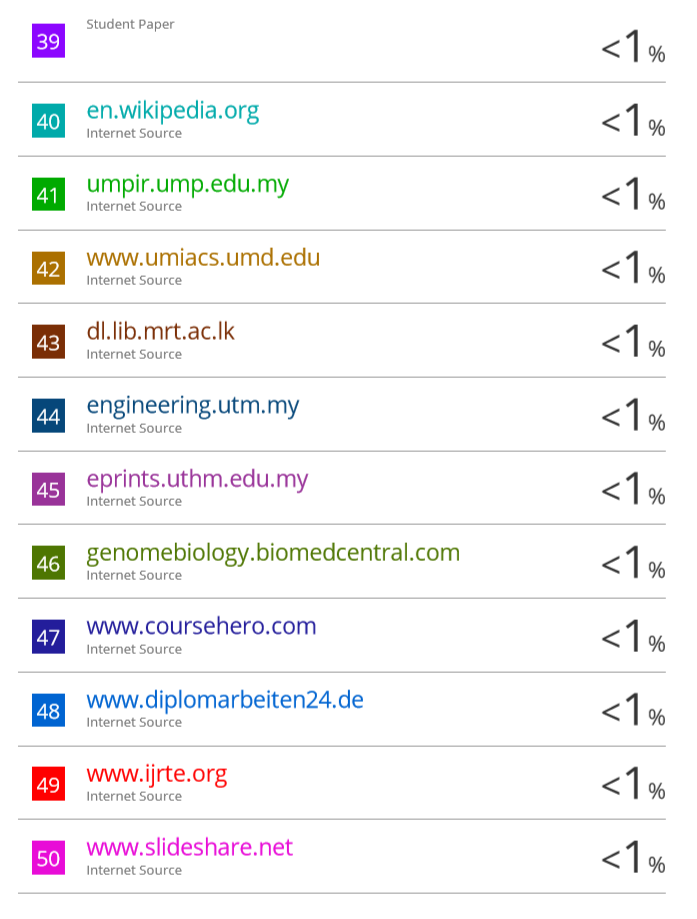
## Appendix - E: Plagiarism Score 03



## Appendix - F: Plagiarism Score 04



## Appendix - G: Plagiarism Score 05



## Appendix - H: Plagiarism Score 06

