KENYATTA UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTING & INFORMATION TECHNOLOGY

IMPROVING EFFECTIVENESS OF INDUSTRIAL PLACEMENT EXPERIENCE USING A RECOMMENDER SYSTEM

DANIEL MULINGE NDOLO

J57/37946/2016

This research project is submitted for the partial fulfillment of the requirements for the award of the degree of Masters of Science in Computer Science in the School of Engineering and Technology of Kenyatta University

2nd February, 2023

Declaration

I declare that this project is my original work and has not been presented in any other university/institution for consideration of any certification. This research project has been complemented by referenced sources duly acknowledged. Where text, graphics, pictures, figures or tables have been borrowed from other sources, including the internet, these are specifically accredited and references cited using current APA system and in accordance with antiplagiarism regulations.

Signature____ Date: ____________________

Name: Daniel Mulinge Ndolo

Registration number: J57/37946 /2016

Department: Computing and Information Technology

Supervisors' declaration*:*

This project has been submitted for appraisal with our/my approval as University Supervisor(s).

Signature ____Date_________________

Name: Dr. J. Kandiri

Department of Computing and Information Technology

Kenyatta University

Table of Contents

Abstract

Different classifier systems have been developed as a result of technological advancement to replace traditional job-search techniques. Finding locations for their industrial attachments is difficult for students in tertiary institutions. By making this training option, they are restricted to open elements like geographic coverage and a limited understanding of industry players. This study used mixed research methodology to conduct its investigation which included survey research and software development methodologies. A multi-criteria classifier called Placement-Ke was developed to create user profiles that are used to forecast and provide recommendations while matching a student with the right company for their industrial attachment training. The classifier was based on the hybrid collaborative filtering algorithm. The recommendations generated were anchored on the user profiled interest and weighted ratings. Using a preexisting data set produced by Kaggle, a base model was created using a pre-trained base model. The base model's accuracy level was 92%. The base model was used to create parameters that were applied in the development of a custom placement recommender system. The model was implemented to a prototype application, which was evaluated using actual users and data. The user comments were recorded on a survey form and examined. According to the analysis, 77.78% of users were happy with the system's overall performance. The inquiry used in the literature evaluation revealed a need for automation in industrial attachment placements to increase their efficiency and speed. The qualitative and quantitative objectives that guided the system design were both met by the research design. The system designed achieved a high user satisfaction rate. This study suggests that the system be developed and implemented for use in higher education institutions. Future work extensions are also discussed.

Keywords: *Recommender engines, filtering technique, mixed hybridization, job recommender systems, industrial placements, machine learning, algorithms and data structures.*

CHAPTER ONE:
INTRODUCTION AND BACKGROUND TO THE STUDY

1.1. Introduction

Decision-making processes are majorly influenced by the form, nature and quality of information we receive concerning certain matters (Awani, 2018). The techno-evolution era devolved this function to digital platforms which are used in collecting individual opinions and perceptions about everyday life consumables (Beniamin, 2017). During this evolution, recommender systems, also known as recommender engines, were created (Webmaster, 2015; Khushee Singh, 2017). These systems have been tested and seen to provide a platform for users to make informed decisions on what to buy, business processes to adopt, choice of friends, jobs to apply for and places to visit. Recommender systems extensively capture data using certain criterion based on the nature of the engine (Attarde & Prof. Manmohan, 2017). The growth of the internet connectivity coverage has hyped the availability of recommender systems to even more users (Nyamwange, 2016). The growth and development of artificial intelligence systems has further catapulted this significant system (KNBS, 2013). Many of these systems include websites used daily such as LinkedIn, Glassdoor, Indeed, CareerBuilder, Dice, Google for jobs, and among others. The LinkedIn Recommender is among the most popular jobs sites and it carries a ‘Talent Match’ feature used by potential employers in identifying top job applicants (LinkedIN, 2019). This is not the only significant feature as the recommender also uses a merger of collaborative filtering and content matching in providing suggestions to jobseekers on potential employers (Rahul , Kulvinder, & Sanjeev, 2014). Eventually, the recommender acts as a personal assistant to the user in terms of finding possible job vacancies and decision making on which jobs to apply for. Linking Industry with Academia is an area of interesting emerging topics which has generated sparking interests in how students get industrial attachments placements (Mc'Odongo, 2019). Higher institutions of learning have a requirement for each student to go through industrial attachment as a way of exposing them to the present job market (Kiplagat, Khamasi, & Karei, 2016). The requirement is gradable where the performance affects the cumulative score of the student before graduation. Getting these opportunities is now a menace since students have to single handedly search and apply for such opportunities (Kiplagat, Khamasi, & Karei, 2016; Mc'Odongo, 2019). Students have to find a way of identifying which industry player suits them best in terms of providing a hands-on learning experience within the specified duration. Owing to the success in jobs seeking through recommender systems, this research seeks to adopt these techniques to aid in finding industrial attachment platforms for students. A dynamic student industrial placement recommender makes it easier for students to search, research, identify, apply, get interviewed and selected within a very short span. Just like other e-recruitment platforms, personalized student industrial recommenders will go a long way in matching students with potential industry players (Mc'Odongo, 2019). The placement system cuts down the hustles and frustrations encountered by students in the process of finding industrial attachment places. The results of the industrial attachment experience will be recorded through ratings and detailed reviews which will be provided by the student after the period of attachment is over. In a survey on job recommender systems, it was identified that artificial intelligence and machine learning techniques have outweighed the traditional Boolean algorithms which makes them faster and convenient in providing real-time results (Shaha & Mourad, 2012).

This chapter provides an outline of the background information of this research work done through a descriptive breakdown of the genesis of the research idea in concert with the general and specific objectives of this research. Also defined is why development of a recommender system and its justification on its application in student industrial placements. This chapter also provides a significance of this research study.

1.2. Background of the study

Industrial attachment in Kenya is a well-structured period of earning where the student gets as much hands-on skills in their areas of study (Kiplagat, Khamasi, & Karei, 2016). According to Nyamwange(2016), apart from the 1st years orientation talk, the students only have an exploration of their careers during the industrial attachment process. Majority of the organizations have resulted in identifying specializations for different job positions in their companies. These positions are associated with key resource of functions of the company. (Muthoni, Gunga, Mutahi, & Origa, 2016). The specializations are meant to tailor those job positions to achieving specific objectives of the company. Kenyan curriculum developers therefore customize majority of university level trainings to include an industry experience where the student is evaluated and assessed afterwards (Nyamwange, 2016). Majority of the Students who undergo appropriate industrial attachment programmes end up developing to be better in terms of creativity and innovation (Muchemi, Muthoni, Mutahi, Gunga, & Origa, 2013). According to Muchemi Et Al(2013), once they resume their studies after the industrial attachment period, majority of the students come up with innovative projects that contribute to their excellence in the industry. These students end up contributing to the development and achievement of vision 2030 in Kenya.

1.3. Statement of the problem

Techno-evolution has seen the introduction of different classifier systems to replace conventional methods of jobs seeking. Such are the multi criteria recommender systems which have seen the automation of e-recruitment processes and medium enterprise and corporate organizations. The advancement of Web 2.0 technology has contributed immensely to an increase in jobs seekers information and e-recruitment enterprise systems (Zheng , Hong, Zhang , & Yang , 2012). Students in tertiary institutions go through a hard time to find places to undertake their industrial attachment (Mc'Odongo, 2019). Selecting a choice for this training limits them to open factors such as geographical coverage and limited knowledge of industry players. According to Zheng Et At (2012), existing job recommender systems adopt the user clustering technique with a single criterion being the primary key for classification. The KNN method was used in this study because it does not need to be trained before producing predictions, allowing for the smooth addition of fresh data without affecting the system's accuracy (Tanya, Shikha, & Dr. Atul, 2018) Tanya et al. (2018) claim that the algorithm gives users the freedom to generate recommendations from a variety of categories. The algorithm has currently been adopted in single criterion environments which has proved successful in job recommendations on different job portals (Sonu Mishra, 2016;Webmaster, 2015). It is this success in other sectors that this research aims to replicate but in an environment with multiple parameters to capture multi-categories in the dataset. The endeavor here is to see the efficacy can be replicated in an industrial attachment environment in a bid to improve the placement process experience.

1.4. Objectives

1.4.1. Purpose of the study

This research project implements Placement-ke, a multi criteria classifier, in building user profiles which will be used in providing predictions and suggestions to a student with an appropriate industry player for their industrial attachment training.

1.4.2. Specific Objectives

1. To investigate the different recommender classifier technologies used in industrial attachment placements and job matching.
2. To design an interactive prototype for industrial attachment placements using multicriteria user clustering and profiling.
3. To develop and implement a prototype industrial attachment placement recommender.
4. To test the prototype using real data and evaluate with actual users.

1.5. Research Questions

1. How have different recommender classifiers been used in industrial attachment placements and job matching?
2. What components of the design will make the multi criteria recommender system achieve user clustering and profiling?
3. How will the prototype implement the multi criteria user profiling and what will be the effect in industrial attachment placements?
4. How will the prototype perform when tested on real data?

1.6. Justification

Recommender systems have been successful in providing suggestions and predictions to users (Shaha & Mourad, 2012). With the ever-increasing dependency on technology and development of affordable technical gadgets, recommender systems have been extensively used in providing quick knowledge-based solutions to many users. Recommender systems have contributed greatly to the development of e-recruitment systems which have been successful (Rahul , Kulvinder, & Sanjeev, 2014). This project provides recommender systems as a solution to industrial attachment placements. Industrial attachment is a well-structured component of a university students' curriculum. Every student is expected to engage in industrial attachment, which is also known as industrial training. The training is evaluated and awarded credit points. The overall points credit to the overall score of the student which determines the possibility of graduating or not. The main purpose of engaging with industry-based experience before graduation is to enable them have a hands-on experience and skills relevant to their area of study.

In a case of public university students, forty-eight percent of students in institutions of higher learning seek for industrial attachment on their own as reported in Kiplagat Et Al (Kiplagat, Khamasi, & Karei, 2016). The project focus tackles this gap by implementing a multi-criteria user clustering algorithm in providing a recommender system that will be a solution to this challenge. Multicriteria user clustering has been successful in providing suggestions to e-commerce online shops such as jumia.co.ke. Owing to this success, this research implements multicriteria user clustering in an effort to enhance the experience of industrial attachment placements.

1.7. Scope

This research proposes the implementation of multi-criteria user clustering in development of an industrial attachment placement recommender system. The prototype developed is not based on a new algorithm but rather implements the already existing algorithm used in hybrid recommenders. The multi-criteria user clustering algorithm works perfectly in providing descriptive analysis and predictions.

1.8. Significance of the study (Rationale)

Students at higher education institutions are one social group that benefits from the effective completion of this research endeavor. Students will be able to access the recommender and identify organizations, companies and institutions that have industrial attachment opportunities. Students will have the opportunity to preview, apply and even communicate with potential training centers all from one platform. Organizations will also have the opportunity to capture some of the best students and potential employees.

CHAPTER TWO
LITERATURE REVIEW

2.1.Introduction

The structural and architectural designs of existing recommender systems are thoroughly examined in this chapter's introduction to recommender systems. The use of recommender systems in corporate digital hiring processes is among the reviewed application areas. The method of placing industrial attachments is highlighted in the chapter, emphasizing the status overview, obstacles, and case study. A suggestion strategy is developed after describing the theoretical and conceptual underpinnings.

2.2.Recommender engines

The idea and use of recommender engines have sparked the interest of many researchers and service providers over the years. These engines have significantly impacted the consumer market, making it easier for them to make decisions based on evaluated predictions and suggestions. (Teng & Lee, 2007). The recommender technology has been applied in various areas, such as recommending movies, blog news articles and e-commerce systems. The growth and development of internet technologies have influenced the development of recommender engines.

As the Internet has become an increasingly important part of our lives, users now face more decision-making challenges (Shahab, Jamshed, & Rashid, 2017). Recommenders have now brought in solutions to the problem-rich research areas. In this case, Teng et al. have defined the recommendation problem as follows (Teng & Lee, 2007). Let *C* be the set of all users, and let *S* be the set of all items to be recommended, such as movies, books, jobs and restaurants.

According to Teng & Lee (2007), the recommendation problem is to choose $S_{ci}, \in S$ whereas its $C_i \in C$ maximizes the utility for each user. As such, $\forall C_i \in C$, S_{ci} = arg max u(c_i, s) where ṳ is the utility function that evaluates the usefulness of the item *s* to the user *c*. This means $u: C \times S \rightarrow R$ where R is an ordered set. However, designing such an approach poses a significant challenge to researchers.

2.3. Implementation of recommender systems in the employment process

The implementation of recommender systems in the employment process involves using data-driven algorithms to make personalized recommendations to job seekers and recruiters. The process starts with the collection of relevant data on job seekers' skills, experience, preferences, and job search history, as well as data on job requirements, previous hiring patterns, and other relevant information. The next step is to choose the appropriate recommender system algorithm based on the nature of the data and the desired outcome. Once the algorithm is selected, the recommender system is trained using the collected data, and its performance is validated through testing and fine-tuning. After successful validation, the recommender system can be deployed into the employment process, providing recommendations to job seekers and recruiters. It is important to regularly update and maintain the recommender system to ensure its accuracy and effectiveness in providing useful and relevant recommendations. This section covers several job suggestion sections, e-recruitment approaches, job recommendation system requirements, and the industrial attachment procedure.

2.3.1. Job Recommendation Systems

According to Al-Otaibi and Ykhlef (2012), research on job recommendation systems has been catapulted by increasing demand for Intelligent Recruitment systems(IRS). Every organization intends to recruit the best candidate for any position suited for them. The ever-growing internet has positioned the global market centrally to allow international communication. This aspect provides a greater variety of candidate options to potential employers. Companies have established e-recruitment platforms that are riding on intelligence that succumbed to recommender systems.

2.3.2. E-recruitment techniques

The company publishes job advertisements in their recruitment portals. On the other hand, job seekers and potential applicants access the system and create user profiles which provide the information needed for the job position they have applied for (Shaha & Mourad, 2012). The profile is stored in the company system server and is used in the future for any application intended to be done. This common pooling of user profiles allows the company to select and shortlist potential candidates based on the job description and requirements. This last functionality gives the company a uniform view of the applicants. According to Shahab et al. (2013), traditional e-recruitment platforms use Boolean search and filtering techniques that do not sufficiently capture the intended level of accuracy in terms of data generation for consideration. Job recommender systems have now come in to provide solutions that range from person-job-fit complexities to selection decisions. The recommender techniques also influence user choice based on suggestions offered. Shaha et al. (2012) determined different factors considered in selection, including the user personality, skillset, and mental abilities. According to their work, these attributes contribute to the personal service delivery and office interrelation traits. From their perspective, these attributes distinguish the following types of persons: -

i. A person who fits in a team.
ii. A person who fits in a job.
iii. A person who fits in an organization.

2.3.3. Job Recommendation system requirements

Various works of literature have been formulated to define the systems requirements for a candidate's job recommendation (Zheng, Hong, Zhang, & Yang, 2012). These include:

i. Job matching should be pegged on skills possessed by applicants.

ii. The system recruitment should factor in both the company and applicant user preferences.

iii. Relational aspects forecast an excellent working relationship between the potential candidates and team members.

iv. People are considered unique and cannot be chosen like books or movies.

In job recommendations, Shaha et al. (2012) suggest that solutions are bidirectional between the job and job seeker (Shaha & Mourad, 2012). According to their research, the recommendation process is divided into two sections: 1) job seeker and 2) job recommendation. Job-seekers can find the highest matching job based on their qualifications and skills. Job recommendations are made to candidates with the best-fitting user profiles. Similarly, Zheng et al. (2012) suggest in their work that the ranking methodology can be simplified to mean that either the top *n* candidates are considered the best fit for the job or the top *n* job profiles are considered for the recommendation (Zheng, Hong, Zhang, & Yang, 2012). In their research, Zheng et al. (2012) outlined the factors to consider, including requirements and preferences. These two factors contribute highly to the job-matching process. Below is a Figure illustrating the model of job recommendation system requirements (Shaha & Mourad, 2012).

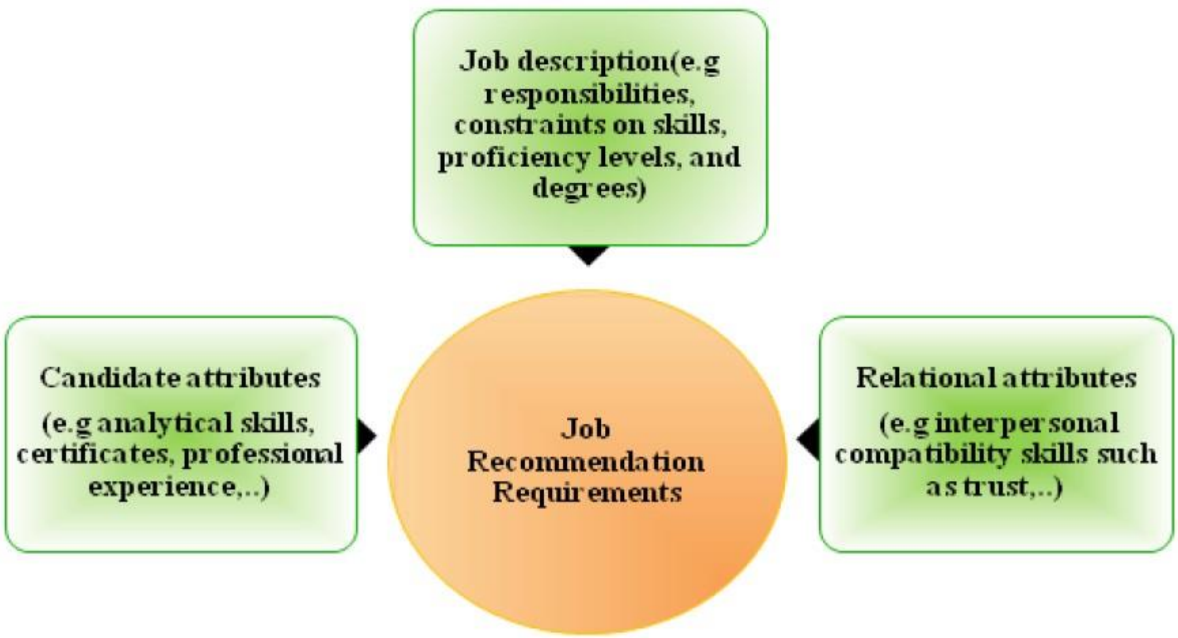


Figure 1: Model of job recommendation system requirements.

This research adopts existing job recommender systems for improvement using the multi-criteria rating algorithm for student industrial placements. In this case, it is essential to understand why this research focuses on industrial attachment placements.

2.4. Recruiting process: Industrial Attachment Placements

During a study on the employability of fashion and garment-making students, it was discovered that there is no coordination in how industrial attachment is handled in Kenya (Kamal, Wamutitu, & Mbugua., 2013). The research showed that most of the graduates only had an excellent mastery of theoretical knowledge, which sometimes was not aligned with the current occupational methodologies. In most cases, students often found themselves settling for any place to get industrial experience basically as a norm for fulfilling graduation requirements in their respective institutions caused of the uncoordinated attachment placements (Kamal et al., 2013; Kiplagat et al., 2016). The disparity in theoretical knowledge and industry trends led to the student's inability to function well. Thus, most companies opted for students who have undergone technical and vocational training.

Example 1: University Student Industrial Attachment Placement in Kenya

Each university has an industrial liaison officer whose partial responsibility is to issue the student an introductory letter and risk insurance cover a period before the student goes for industrial attachment (SU, 2017; Capital Campus, 2014). After this, the students are allowed to look for industrial attachment centers on their own. After the student identifies a potential training place, they apply and attach the documents above to prove they are students with the said institutions. For the less informed and exposed students, this becomes a challenge, and thus they are fixed at the point of joining any firm around them. According to an informal article written by a case example student, Nixon Awani, high-end corporate organizations rarely pick random students for industrial attachment in Kenya (Awani, 2018). There is a tendency for the company leadership to engage students deemed to be known by staff members such as distant relatives and family friends.

Example 2: Student Industrial Attachment in Vocational Centers

Vocational Training centers such as Kenya Medical Training College conduct the placement on behalf of the students, making them develop partnerships relevant to the field of study (KMTC, 2018). This cuts the hustle for the student to go around in selecting and applying for these attachment opportunities. It ends up saving time, money and energy for the student and chances of getting more practical skills are higher.

A Case Study: Kenya University Industrial Attachment Placement

A study was conducted at the University of Eldoret on how students got industrial attachment centers (Kiplagat, Khamasi, & Karei, 2016). 48% of the students acknowledged that they got attachment places by engaging in direct applications to the company, while 26% got the placement through close friends (Kiplagat, Khamasi, & Karei, 2016). 9% of the students got the placement through parents and relatives, while only 3% got placements in organizations they previously worked (Kiplagat, Khamasi, & Karei, 2016). 0% of the students got a placement through the respective departments in their institutions.

Some of the students' recommendations were for the academic departments to participate more in helping them get places for attachment. Others suggested a consolidated forum where students can select places for attachment.

Following this recommendation, this research project is appropriate and solves a big problem connecting students through an online platform. If 48% of the students managed to get an attachment by applying directly to the companies, it means that this number can grow even much more if there is an open and consolidated center for students and companies
to interact.

Through the industrial Training Act(Amended) of 2011, the Government of Kenya established a National Industrial Training Authority whose partial role was to facilitate linking between companies and students looking for places to be attached (NITA, 2018). The Authority took up the responsibility by creating intentional partnerships with organizations and institutions that offer industrial attachment opportunities. The corporation also developed an in-house industrial attachment portal for placements dubbed ITAP. The portal allows students to view and search for potential companies while the companies can post any vacancies they have for the students to view (NITA, 2018). Institutions of Learning Industrial Liaisons Officers have also had the opportunity to create profiles. The system has had shortcomings since there have been significant breakdowns for a while. So far, the corporation has only linked approximately 15,000 students to industrial attachment centers.

The other platforms used to advertise for industrial attachment places include renowned job recruitment portals such as Brightermonday. Co.ke (BrighterMonday, 2018). The companies or organizations sought to have students for industrial attachment pay to have classified ads in such job recruitment portals. These job recruitment portals majorly favour graduates since they provide job and internship opportunities. Only a few organizations advertise industrial attachment opportunities (BrighterMonday, 2018).

2.5.Classification of Recommender Systems

Shahab et al. (2017) analyzed the different recommender systems based on the approach used, the area where the recommendation is made, and the data mining techniques, among others. These types of recommenders fall into five broad classification areas. The classification areas are content-based, knowledge-based, collaborative, utility-based, and demographic-based recommendations. According to Shahab et al. (2017) and Wenxing et al. (2013), the filtering algorithm makes these recommendation categories unique. The various algorithms include collaborative filtering, demographic filtering and content-based filtering techniques.

The researchers have broadly categorized recommender systems into eight types that cover the new generation of recommender systems (Shahab, Jamshed, & Rashid, 2017). Collaborative filtering-based recommender systems, reclusive methods-based recommender systems, demographic filtering-based recommender systems, knowledge-based recommender systems, hybrid recommender systems, context-aware recommendation systems, social network-based recommendation systems, and soft computing techniques-based recommender systems are some of the types (Wenxing Hong, 2013; Shaha & Mourad, 2012; Shahab, Jamshed, & Rashid, 2017).

2.6.Characteristics and Challenges of filtering techniques:

Based on research findings from Shahab et al. (2017) and Al-Otaibi & Ykhlef(2012), it is evident that the basis for choosing or designing a recommender algorithm is dependent on three major factors, namely:

i. Proposed Area of Application of the targeted recommendation

ii. The recommended approach to be used

iii. Intended data mining technique to be applied.

Their research has classified the three major recommendation approaches based on their characteristics and challenges. In Table 2.1 below, the researchers have classified the characteristics and challenges of content-based filtering, collaborative filtering and knowledge-based filtering approaches.

Recommendation approaches	Characteristics	Challenges
Collaborative filtering	• Independent of the proposed items' machine-readable representations. • Work well with intricate elements like music and videos. • Domain expertise is not required. • Quality becomes better with time.	• Ramp-up issues with new users and things. • Performance suffered when user-item matrix grew in size. • limited scaling for large amounts of data. • Model construction with model-based approaches is expensive.
Content-based filtering	• Domain expertise is not required. • When express rating is difficult, work well with implicit feedback. • Quality becomes better with time.	• Problem with ramp-up for new users. • The characteristics linked to recommended objects have performance restrictions.
Knowledge-based	• There is no need to learn more about a specific user because the system's assessments are not influenced by personal preferences. • Zero ramp-up issues.	• Acquiring knowledge is necessary. • Challenges with knowledge engineering.

Table 2. 1Characteristics and challenges of the 3 recommender system approaches

2.7.Existing recommender systems

In order to understand the prototype architecture, this research looks into the existing best-fit job recommender systems that can be improved. As discussed earlier, Shaha et al. (2012) present a summarized taxonomy of job recommender systems (Shaha & Mourad, 2012). This taxonomy illustrates the current recommendation approaches for job matching. Shahab et al. (2017) extended the taxonomical model, which has clearly outlined the personnel e-recruitment election approach.

To investigate more on the current types of hybrid recommender systems, we look into the approaches described, including 1) Hybrid job recommender systems according to (Shaha & Mourad (2012).) and 2) Hybrid recommender systems according to (Tanya, Shikha, & Dr. Atul, 2018).

2.7.1. Types of Hybrid job recommender systems

One type of hybrid system is the proactive job recommender system. Utilizing hypermedia, proactive recommender systems enhanced existing recommender systems (Shaha & Mourad, 2012). Essential parts of the system include an ontology checker, web spider, profile analyzer, preference analyzer, and user interface generator. The web spider is responsible for finding and gathering job-related data from external sources. The information is passed to the ontology checker, which matches the data with various ontologies and then performs classification. The raw job data is stored in a predesignated form. Once stored, the profile analyzer publishes the recommendations to the user using the weighted nested group of user favorites; by comparing the differences in weight with any job openings. Based on this performance, the system can generate a list of recommended jobs. Finally, the preference analyzer matches the system listing with the user preferences.

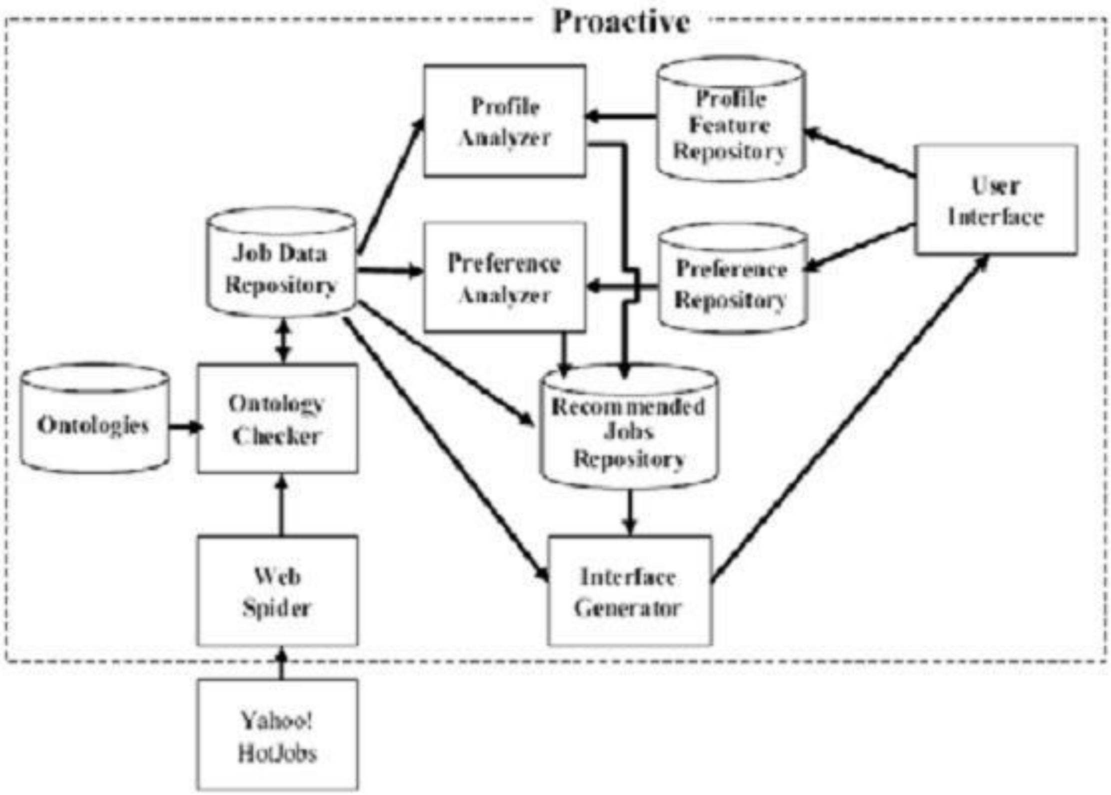


Figure 2.2: Model of proactive job recommender system

According to Wenxing et al. (2017), the probabilistic hybrid approach approach is common in movie recommender systems. The approach encompasses both content-based and collaborative filtering simultaneously. This concept solves the issue of data sparsity. This hybrid approach adopts the value of each user in object pairs (x, y), where y is the set of objects, and x is the set of users. The expectation-maximization algorithm determines the parameters of this model. The aim is to produce a rating matrix that determines the recommendation to the user based on the assessed values from the candidate's profile with a probability of getting an accurate match (Shahab, Jamshed, & Rashid, 2017; Shaha & Mourad, 2012).

An example cited by Wenxing et al. (2017) proposed applying this approach by developing two separate recommender engines. One would be a job recommender, while the other would capture the curriculum vitae of the user. In their work, they developed the cv recommender first and then developed the job recommender in concert (Wenxing, Siting, & Huan, 2013). The cv recommender system would select the best-suited cv for a specific job profile to a recruiter. The second system would capture the user information from their CVs and then recommend the most suitable job to the potential client. This integration created a multilayered system that collaborated with a different perspective from a hybrid approach.

Another type is the fuzzy multi-criteria approach, where the system algorithm performs by determining the candidate's suitability based on specialized skills and personality interests. According to Shaha and Mourad (2012), This approach is achieved through a statistical analysis of user information through an analytic hierarchy process (AHP). The process is a dynamically multi-objective decision-making method encompassing uncertainty in decision-making and other assessment criteria. The AHP analyzes the weight of different factors then predicts based on a linguistic variable and fuzzy number. The final weighted cluster determines the basis of selection. An example is the Absolventen.at Job Recommender, that combines collaborative filtering, content-based filtering and knowledge-based recommendation techniques. The system depends on user feedback to rate the recommendations on salary, location and other preferential results.

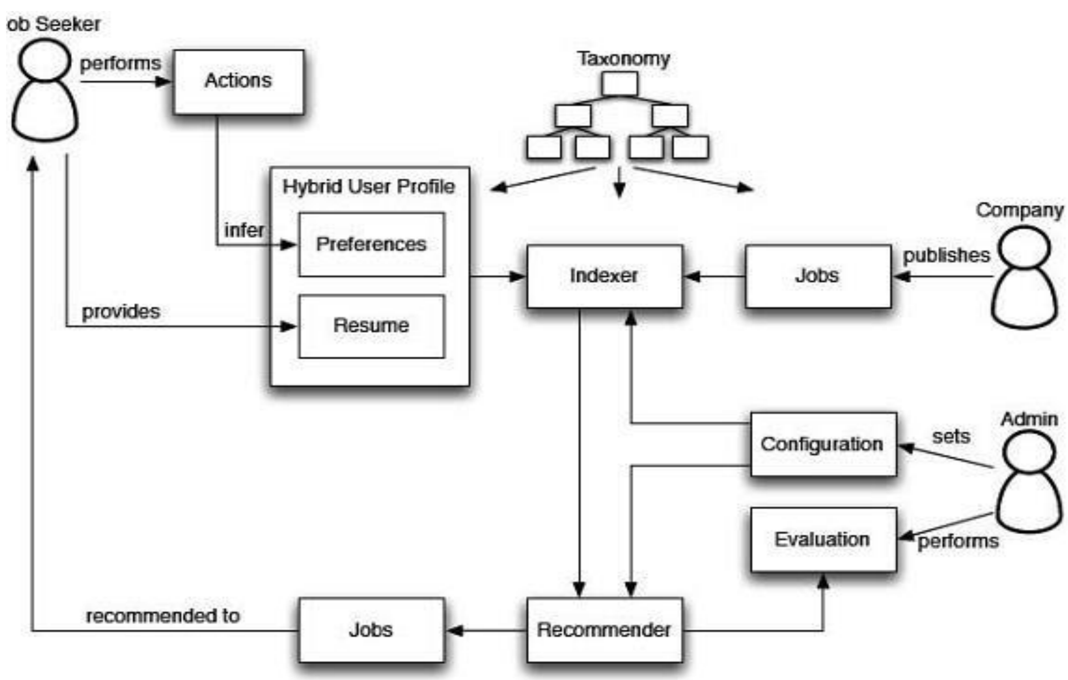


Figure 2. 1 Absolventen.at Job Recommender

Shaha and Mourad (2012) attempted to improve the matching process by combining different strategies in a Semantic matchmaking Approach to create an adaptive job provision and discovery environment. This approach adopts logic-based and similarity-based matching techniques. In the initial stage, they applied a deductive model to determine the match between a candidate and a job offer. In the latter stage, they used the output to calculate the similarity measure to rank the applicants with a partial match (Shaha & Mourad, 2012). The table below summarizes Shaha and Mourad's description of different hybrid recommender systems, showing each technique's advantages and disadvantages.

Techniques	**Advantages**	**Disadvantages**
Probabilistic hybrid approach.	• Cross-directional advice • It includes relational elements..	• Exclusively in binary format. • Less attributes are used. • There aren't any ideal measurements.
Proactive job recommender system.	• Intelligent system. • Use various qualities. • Use ontology as a knowledge base to define features and a classification system for tasks (attenuate cold-start problem).	• Use of keywords in search. • Single-directional advice. • Issues with knowledge engineering and knowledge acquisition. • Relational components are not present.

Semantic matchmaking for job recruitment	• A two-way suggestion • Efficient ways for matching. • Has a wide range of qualities. • Included are features of relationships. • Representation of both quality and quantity (proficiency level for skills is included). • Use two levels for the skills comparison (constraints and preferences).	• Problems with knowledge engineering and knowledge acquisition. • technology and tools not included.
Fuzzy multiple criteria method for recruitment.	• Use various qualities. • Included are features of relationships. • efficient ways for matching. • To ascertain skill levels, consider linguistic variables.	One-way recommendation.

Table 2. 2: Advantages and Disadvantages of different hybrid recommender techniques

2.7.2 Hybrid recommender systems

Based on Tanya et al. (2018) analysis of recommendation systems, we get three types of hybrid systems:

One is the weighted hybrid system. This type of hybrid system involves a combination of a collaborative and content-based filtering algorithm to improve the system's performance. According to Tanya et al. (2018), the system computed the total weighted clusters and compared them to select prediction items. This type of system solves the challenge of running a single algorithm filtering technique in weighted clustering.

Second are the switching hybrid systems. This system is the 'daily learner,' which involves collaborative and content-based filtering. However, the system uses only one filtering in the weighted averaging clustering. The choice of filtering is based on the user's needs. After determination, the system can switch from one filtering method to another to solve the immediate need. The disadvantage of this system is that it cannot apply both filtering methods.

Third are the systems that implement Mixed Hybridization techniques. This hybrid system involves several recommendation techniques and displays different selection recommendations per object. This system improves the single recommendation per object approach by producing several recommendations for one object. However, the disadvantage is that the recommendations are not weighted across the different objects, but each technique has its recommendation. This may confuse the user.

Last are the Cascade hybrid systems. An example of this system is EntreeC, which involves collaborative and knowledge-based recommenders. This system allows the results of one recommender to act as input to other recommendation techniques. This improves the final presented recommendation to the user. A significant advantage of this system is that is tolerant to noise.

2.8. Supervised Learning

Modeling is used to train the computer (Moore, 2017). According to the researcher, the operator provides the machine learning algorithm with a known dataset that contains the necessary inputs and outputs; the system then has to work out how to access those inputs and outputs. The operator is aware of the correct solutions to the issue, even while the algorithm finds patterns in the data, gains knowledge through observations, and makes predictions. The algorithm is adjusted by the operator as predictions are made, and this cycle is continued until the method is effective and very accurate. In supervised learning, classification, regression, and forecasting are all included (Rahul, Kulvinder, & Sanjeev, 2014).

Classification: A machine learning algorithm must draw a conclusion from observed values in classification tasks and determine which group newly observed data belongs to. For example, the computer needs to look at recent observational data and categorize emails as "spam" or "not spam" in line with that data.

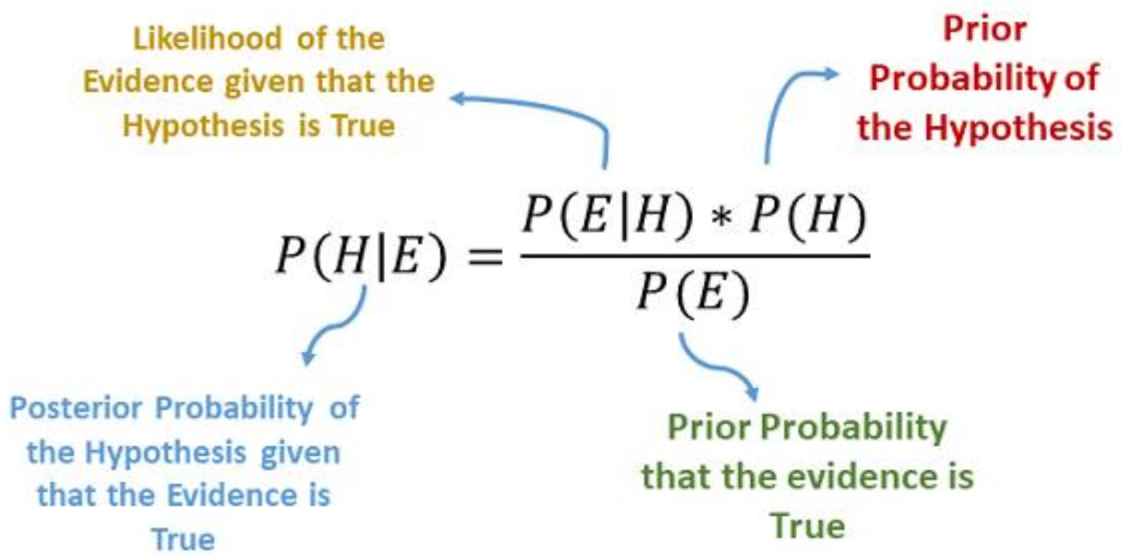
Regression: In regression problems, the machine learning system must estimate and understand the relationships between the variables. Since regression analysis focuses on one dependent variable and a number of other varying factors, it is useful for prediction and forecasting.

Forecasting: Forecasting is a common technique for trend analysis that involves making predictions based on historical and current data.

2.9. Summary of the most prevalent and well-liked machine learning algorithms

Based on studies by Tanya, Shikha, and Dr. Atul (2018), Attard and Prof. Manmohan (2017), Sonu Mishra (2016), Webmaster (2015), and Ykhlef (2012), this study featured a summary of key machine learning algorithms and a summary of how they work.

One is the Naïve Bayes Classifier Algorithm, otherwise termed as the Supervised Learning Classification method. The Naive Bayes classifier classifies each value as independent of every other value based on the Bayes theorem. Using a certain collection of information, it lets us to predict a class or category using probability. The classifier works wonderfully while being straightforward and is commonly used since it outperforms more complicated classification methods.



Second is the Support Vector Machine Algorithm. Techniques for support vector machines analyze the data utilized in regression and classification analysis. They have models for supervised learning. Each set is identified as fitting into one of the two categories by providing training examples, thereby categorizing the data. After that, the algorithm builds a model that assigns new values to one or both categories.

Third is the linear regression. Linear regression is the most basic type of regression. Simple linear regression allows us to understand the associations between two continuous variables.

Fourth is the logistic regression. Calculating the probability of an event given the historical data that is available is the main objective of logistic regression. It addresses a binary dependent variable, which can only take on the values 0 or 1.

Fifth is the decision trees algorithm. The probable outcomes of each action are displayed via branching in a decision tree, a form of tree structure that resembles a flowchart. The results of each test performed on a specific variable are shown on each branch of the tree, which represents a test at each node.

Sixth is the random forests. A sort of ensemble learning called random forests, also referred to as "random decision forests," integrates many methods to provide improved classification, regression, and other task-related results. Although each classifier performs best when used together, they are all ineffective on their own. The "decision tree" of the algorithm, which represents decisions in the shape of a tree, is where input is entered. As it progresses down the tree, the data is subsequently divided into smaller and smaller sets according to particular variables.

Seventh is the Nearest Neighbors algorithm. Using the K-Nearest-Neighbor method, one may determine the likelihood that a given data point belongs to a specific group. To determine which group a particular data point belongs to, it looks at the surrounding data points. As an illustration, if a data point is found on a grid and the algorithm tries to determine which group it belongs to (for example, Group A or Group B), it will look at neighboring data points to determine which group the majority of the points are found in.

2.10. The gap in existing recommender systems

(Attarde & Prof. Manmohan, 2017) Research findings show that existing recommendation systems have issues that may challenge the process of evaluating results and presenting the recommended solutions. The researcher clearly showed that existing recommender systems had adopted a single-criterion methodology. (Shahab, Jamshed, & Rashid, 2017) also agree to cite that the methodology involves the generation of recommendations based on one category in the dataset. This method has several challenges identified by (Tanya, Shikha, & Dr. Atul, 2018). The major challenges cited are highlighted below.

One is the challenge of data sparsity. According to Tanya Et. Al. (2018), in practice, most recommender systems are established on large schemas and datasets. This feature makes the user matrix complex, especially in the user matrix on objects and items. This factor makes the system's performance slow since the recommendation is based on the user's preferences or past feedback. This challenge also comes in recommender systems that depend on ratings, such as movie recommender engines. This is because the newer the user object, the slower the prediction since each product has to be rated substantially before accurate results are delivered.

Second is the Cold Start Challenge. This challenge is visible in e-commerce recommender systems where the predictions will take a long before they begin to deliver a highly commendable result. This is because there is no preexisting user history and the basis of recommendation is user-profiled preferences in unsupervised learning recommenders.

Third is the challenge of system scalability. The complexity of the hybrid algorithms poses a challenge in growing the number of users accessing and depending on recommender systems. However, this challenge has been partially tackled by including user clusters in recommender systems. Large companies such as social media moguls Twitter have had to create and improve cluster machines that manage millions of users.

2.11 Approaches to recommendation systems.

Using a comparative analysis of the literature review, this research identifies gaps in the existing types of recommender systems. As we appreciate work identifying the existing recommender systems, we identify and adopt the hybrid multi-criteria system as the most effective for job recommendations. This research project identifies the challenge of tackling one-way recommendation in fuzzy multi-criteria recommenders by adopting mixed hybridization. According to Tanya et al. (2018), mixed hybridization is the better of all techniques since it involves an average weighted cumulative that determines the predicted recommendation. The mixed hybridization adopts the multi-criterion KNN algorithm, which is able o filter recommendations from two categories in one dataset (Tanya, Shikha, & Dr. Atul, 2018). This aspect solves the one-way recommendation challenge to produce a weighted result in hybrid fuzzy multi-criteria systems. In this research, we conclude that collaborative and content-based filtering algorithms would produce a weighted result informed by the different user preferences. Applying this proposal will be a long way in providing a solution to the student industrial attachment placements. For example, according to Nita (2018), the current system only allows students to apply for industrial attachment based on the organization they know and job feedback. The user does not have a platform the enter their preferences but rather apply through the system. The system does not allow for communication between the company and potential candidates.

2.11. Theoretical Framework

Recommender systems operate based on three aspects: rationale, approach and operation. This research focuses on the approach aspect, which breakdown into a more personalized approach, as shown in the figure below.

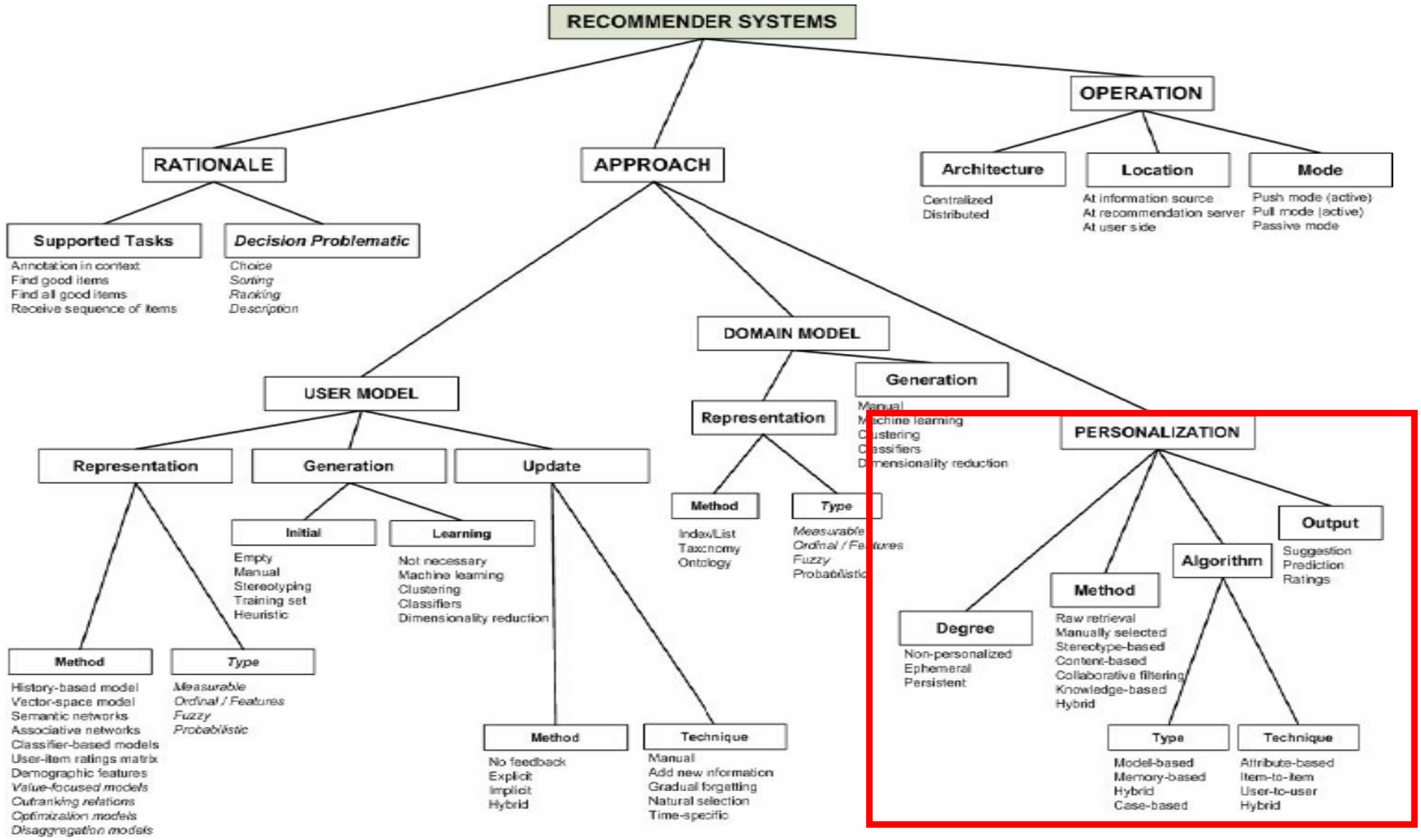


Figure 3:The theoretical framework for the analysis and classification in the recommender system

2.13. Conceptual Framework

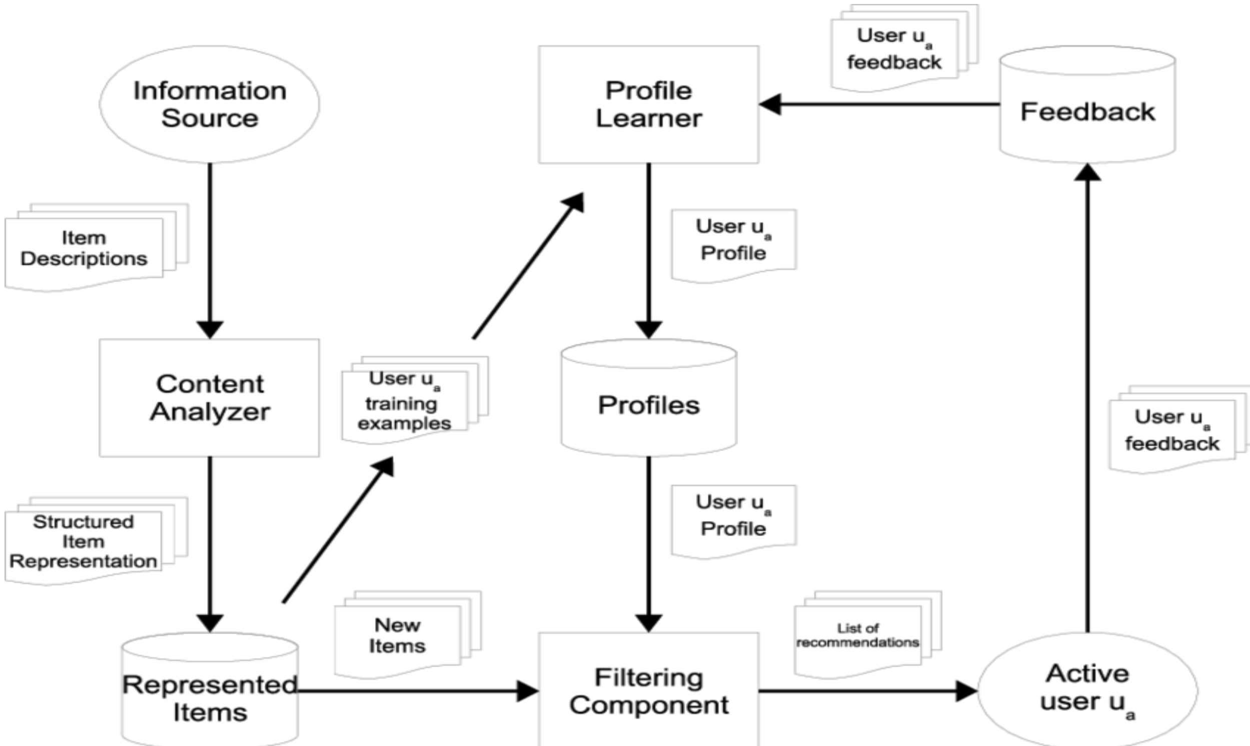


Figure 4: Model design of a hybrid recommender system

CHAPTER THREE:
RESEARCH METHODOLOGY

3.1. Introduction

This chapter outlines the mixed research methodology adopted in the project. This involved review of existing literature in the background study, systems development methodology alled rapid application development, and qualitative methodology for system validation. This involved employing a multi-criteria approach to design, develop, and implement a dynamic recommender system for student industrial attachment placements. The project is qualitative and adheres to the goals of qualitative research. The research design, strategy, model design and execution, and system deployment will be the main topics of this part.

3.2. Research Design

This study adopted a problem-solving paradigm called Design Science Research (DSR) aims to increase human understanding by creating creative products (Zheng, Hong, Zhang, & Yang, 2012). DSR's goal is to create unique artifacts that address problems and improve the contexts in which they are instantiated. This is how it seeks to advance knowledge bases in science and technology. Design science research techniques are used in this study to provide a solution based on the objectives. This strategy works by creating design artifacts that provide social or organizational answers to human problems, hence it has a good application in computer science and information systems (Moore, 2017). It combines mathematical and computational techniques that have a big impact on how well generated artifacts are measured for quality and efficacy. The principles of design, social, natural, and formal sciences serve as the foundation for this research methodology, according to Moore (2017). (Attard & Prof. Manmohan, 2017). In place of the currently used empirical approaches, this work uses proof methods for verification.

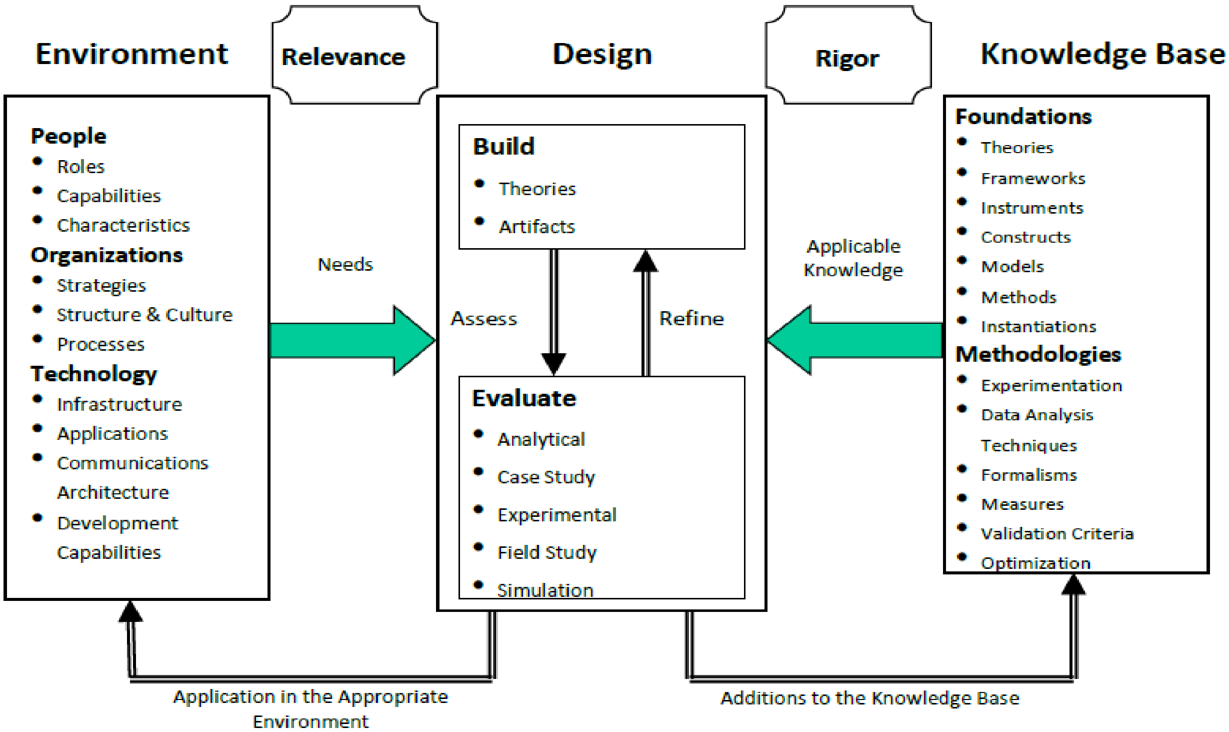


Figure 5: Design Science Research Framework (Adapted from (Hevner et al., 2004))

The design science approach was adopted as the research methodology since its fundamental approach encompasses having knowledge and understanding of the problem of human social organization. The approach also factors in the design and building of designed artifacts (Teng & Lee, 2007). The design science approach has two significant processes: prototype building and evaluation (Zheng, Hong, Zhang, & Yang, 2012). The design science approach employs the rapid application development technique in artifact development.

The success of DSR projects has been based on a variety of process models, including those developed by Nunamaker, Chen, and Purdin in 1991, Walls, Widmeyer, and El Sawy in 1992, Hevner in 2007, and Kuchler and Vaishnavi in 2008. Peffers, et al. created the model that has garnered the most attention (2008).

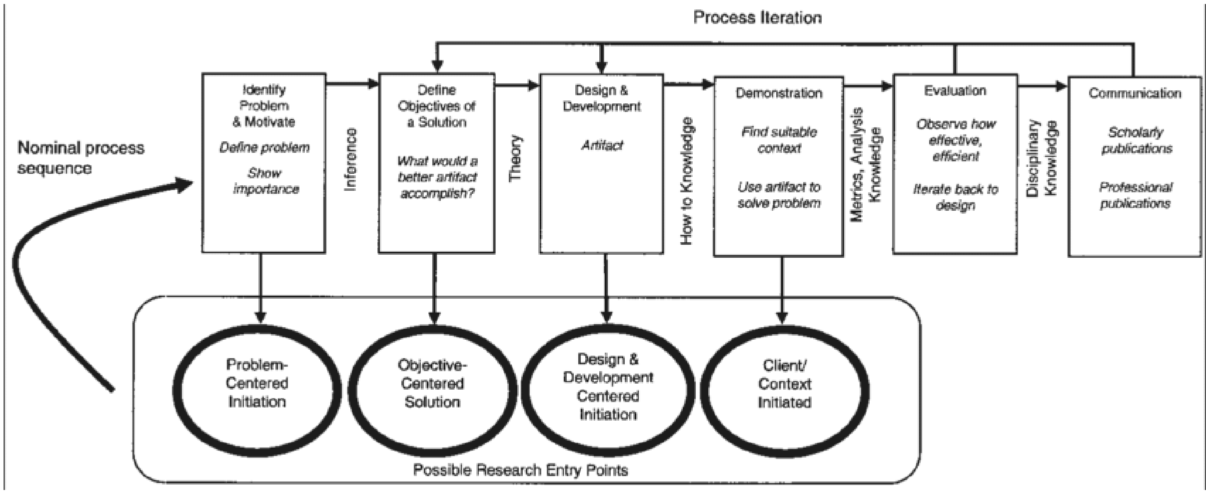


Figure 6:DSR Methodology Process Model (Adapted from Peffers et al. (2008))

The DSR process consists of six steps: problem identification and motivation, setting goals for a solution, design and development, demonstration, assessment, and communication. There are four possible entry points: problem-centered, objective, design and development-centered, and client/context-centered. The following is a brief description of each DSR activity. Below is a synopsis of each DSR activity.

3.2.1. Problem identification and solution objectives.

The project's main objective is to close this gap by applying a multi-criteria user clustering technique to create a recommender system that would address the industrial attachment placements problem. Multicriteria user clustering has effectively offered recommendations to online stores for e-commerce, such as jumia.co.ke and job recommendation engines such as LinkedIn. By utilizing multi-criteria user clustering in a collaborative filtering environment, this study hopes to enhance the industrial attachment placement experience. The solution targets identifying the different parameters that determine a successful placement and creating an artifact to bridge the gaps. According to the design science model, the need lies in identifying the parameters that control the industrial placement process. This research identifies three major areas that need intervention.

One is the need by the student to complete the academic requirements, including industrial training. According to Moore (2017), the pressure to finish and proceed on to completion provokes desperation leading to 60% of the students opting for any relevant attachment location.

The second area is the user-interests generated results. This research finds that the academic requirement objective is for each student to get placed based on the area of study and not necessarily interests.

The third area is on desire to be placed in top-rated organizations, according to Kiplagat Et. Al (2013), students envision to be attached in top-rated organizations on the assumption that they provide a better practical experience.

3.2.2. Design and Development

This stage involves the activities that lead to the design and development of a successful artifact. Software development approaches can be divided into two main types based on the degree of complexity in the development flow structure. Systems range from the extremely complex—referred to as heavyweights by Moore (2017)—to the simpler, lighter methods. While less complex techniques like prototypes, extreme programming, rapid application development, and other agile methods require little to no documentation, more sophisticated techniques like waterfall, spiral, and component-based models can call for extensive documentation. Rapid application development of prototypes was chosen as the method for this study based on its research design in the James Martin Model Omankwu et al. (2018).

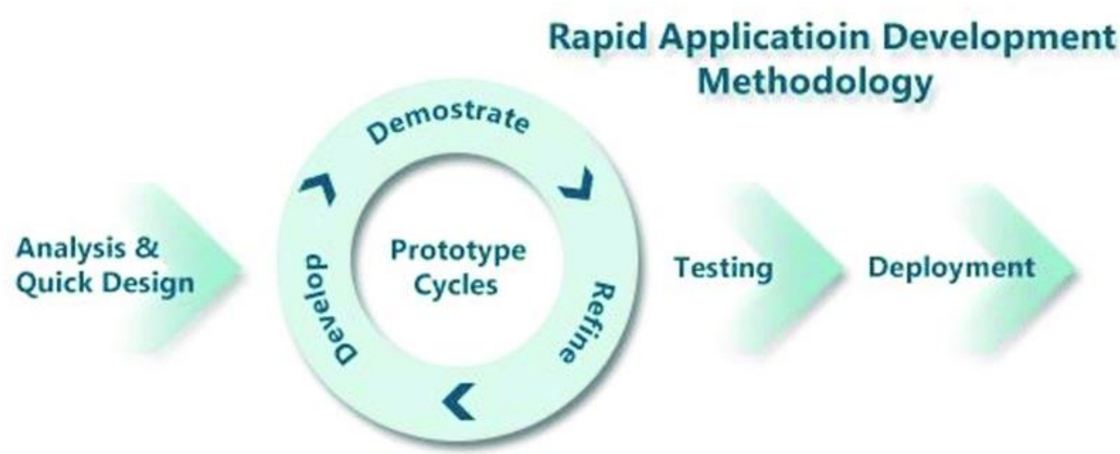


Figure 7: James Martin Rapid Application Development Model Omankwu et al. (2018).

3.2.3. Requirements Analysis

Key attributes used are captured from what is required during the industrial attachment application, as identified in our literature review. These attributes form the required input fields that the recommender system will evaluate.

3.2.4. System Design

The design of the system prototype was based on design modeling methods such as use case diagrams, dataset sequence diagrams, and database class diagrams. A component diagram from the use case diagram had to be described during this stage.

Figure 8: Use-case Diagram

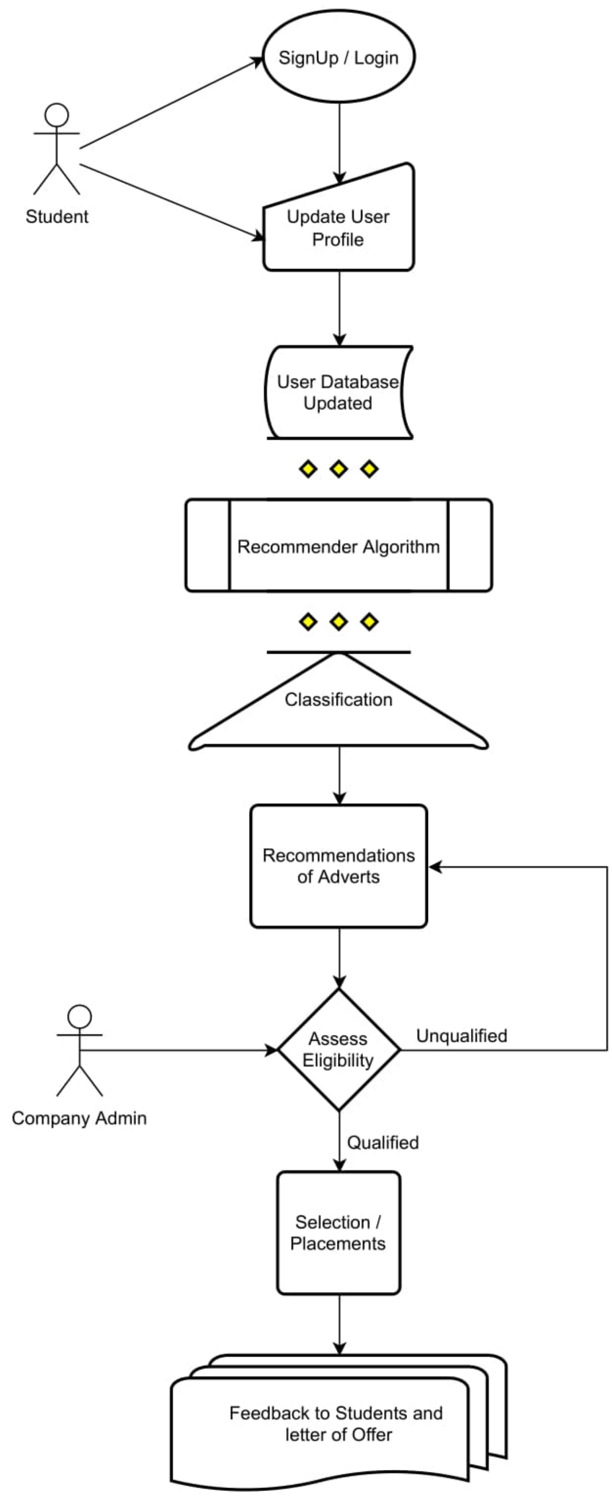


Figure 9: Process Flow

3.2.5. System Development

The prototype was created through a phased system development process. Modeling the system based on the system design was the initial stage. Model loading and deployment made up the second phase. The system prototype was tested and evaluated in the third phase. The third phase was testing and evaluation of the system prototype.

3.3. Model Design and Implementation

This stage involved designing a model that would dictate the system's performance. Due to the system's complexity, the James Martin model of intelligent prototypes proposes using a base model with a large dataset for training a custom model with fairly low entries to help overcome the challenges of custom recommender systems. These challenges include the data disparity and cold-start problem as described in our literature review.

3.3.1. Base Model

The foundational model created was based on an intelligent book recommendation system that was functionally implemented in a top-tier institutional library. The model collected the basic input that would have been weighted in the analysis. The model was pre-trained using a suitable dataset that was found through this research. A collaborative filtering process based on user profiles and the item-based rating was activated by the multi-criteria algorithm used. The two categorization techniques would generate user-relevant recommendations.

3.3.2. Data Exploration

The dataset used in this research model includes information about the books, the authors, and other pertinent details. The dataset was obtained from Kaggle good read book. Primarily, the dataset offers the comprehensive information about the books. There is a detailed description next to each column.

Column	Description
bookID	An individual book identification number
title	The name published with which the book was released
authors	Names of the book's authors. The delimiter "many authors" is used to
average_rating	The overall rating for the book was on the average

isbn	The International Standard Book Number is yet another special identifier for the book.
isbn13	A 13-digit ISBN rather than the usual 11-digit ISBN to identify the book
language_code	Aids in comprehension of the book's main language. For example, eng is the accepted form of English.
num_pages	Number of pages in the book
ratings_count	Overall rating total for the book.
text_reviews_count	Total amount of book reviews that were written

Once the data was loaded, the next step was to locate each and every null value in our data. Missing data makes the data murkier. It is shown as NA or NAN. Every data point is significant when the dataset is small. The absence of data causes an imbalance in the observations and may even result in incorrect conclusions.

```
df.isnull().sum(), df.describe()
```

The code above was used in the checking of null values and mean scores of ratings done on the books.

3.3.2.1. Data Classification and visualization

The data loaded was classified into a standard format arrangement to ensure all the books had a standard rating. As a way of testing the data, we visualized the data based on different parameters. These parameters would determine the performance of the algorithm in generating quality results. The data visualization showed the top authors' information according to the number of books they had published. The data visualization also included the most frequently reviewed books based on how often they were rated. Another relation identified was between the average score and the total number of scores. The distribution of average scores with a book's page count, the language used in the book, and the quantity of text reviews were among the other relationships that were graphically represented.

```python
top_ten = df[df['ratings_count'] > 1000000]
top_ten.sort_values(by='average_rating', ascending=False)
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(10, 10))
data = top_ten.sort_values(by='average_rating', ascending=False).head(10)
sns.barplot(x="average_rating", y="title", data=data, palette='inferno')
```

Code to view top authors

```python
most_rated = df.sort_values('ratings_count', ascending = False).head(10).set_index('title')
plt.figure(figsize=(15,10))
ax = sns.barplot(most_rated['ratings_count'], most_rated.index, palette = 'inferno')
totals = []
for i in ax.patches:
    totals.append(i.get_width())
total = sum(totals)
for i in ax.patches:
    ax.text(i.get_width()+.2, i.get_y()+.2,str(round(i.get_width())), fontsize=15,color='black')
plt.show()
```

Code to view most reviewed books

```python
df.average_rating = df.average_rating.astype(float)
fig, ax = plt.subplots(figsize=[15,10])
sns.distplot(df['average_rating'],ax=ax)
ax.set_title('Average rating distribution for all books',fontsize=20)
ax.set_xlabel('Average rating',fontsize=13)
```

Code to view average rating distribution

```python
ax = sns.relplot(data=df, x="average_rating", y="ratings_count", color = 'red', sizes=(100, 200), height=7, marker='o')
plt.title("Relation between Rating counts and Average Ratings",fontsize = 15)
ax.set_axis_labels("Average Rating", "Ratings Count")
```

Code to view Rating counts and Average Ratings

3.3.2.2. Data Preparation

Ranges between various categories are now provided in a new column called "rating between." This functioned as one of the features that were incorporated into our model to improve predictions. Two new Data Frames containing the different values of the rating_between column were created. If a grade falls into a certain category, such as grades 4 and 5, assign a value of 1, while the other grades will be given a value of 0.

```
df2.loc[ (df2['average_rating'] >= 0) & (df2['average_rating'] <= 1), 'rating_between'] = "between 0 and 1"
df2.loc[ (df2['average_rating'] > 1) & (df2['average_rating'] <= 2), 'rating_between'] = "between 1 and 2"
df2.loc[ (df2['average_rating'] > 2) & (df2['average_rating'] <= 3), 'rating_between'] = "between 2 and 3"
df2.loc[ (df2['average_rating'] > 3) & (df2['average_rating'] <= 4), 'rating_between'] = "between 3 and 4"
df2.loc[ (df2['average_rating'] > 4) & (df2['average_rating'] <= 5), 'rating_between'] = "between 4 and 5"
```

The language code column was split in half to retrieve each language separately and assign them values of 1 and 0, where 1 was assigned if the book was written in a specific language, such as English, and 0 if it wasn't.

```
rating_df = pd.get_dummies(df2['rating_between'])
language_df = pd.get_dummies(df2['language_code'])
```

These two data frames were concatenated into one and named features. The intelligent book recommendation system could use the capability provided by this DataFrame. Rating_df and language_df values were present in it. The dataframe also included average grade and number of grades variables.

```
features = pd.concat([rating_df,
                      language_df,
                      df2['average_rating'],
                      df2['ratings_count']], axis=1)
```

3.3.2.3. The recommendation algorithm

Today, recommender systems power the majority of the online products we use. Websites like Youtube, Netflix, Amazon, Pinterest, and a list of other online services rely on recommender systems to sort through millions of pieces of material and provide their customers with individualized recommendations. It has been well-established that recommender systems offer both internet firms and their customers enormous benefits. In 2009, Netflix gave a development team a $1 million reward for creating an algorithm that improved the company's recommendation system by 10% (Liao, 2018). The KNN algorithm was used in this work to weigh the average rating. KNN depends on item feature similarity rather than making any assumptions about the underlying data distribution.

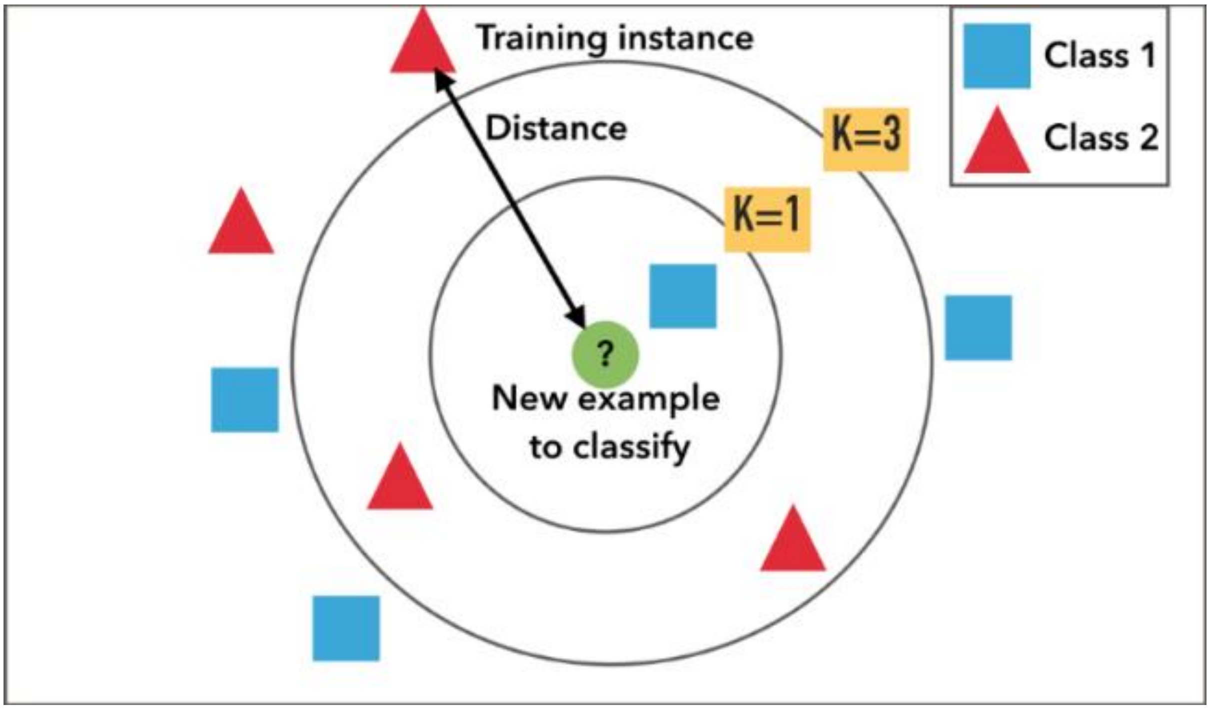
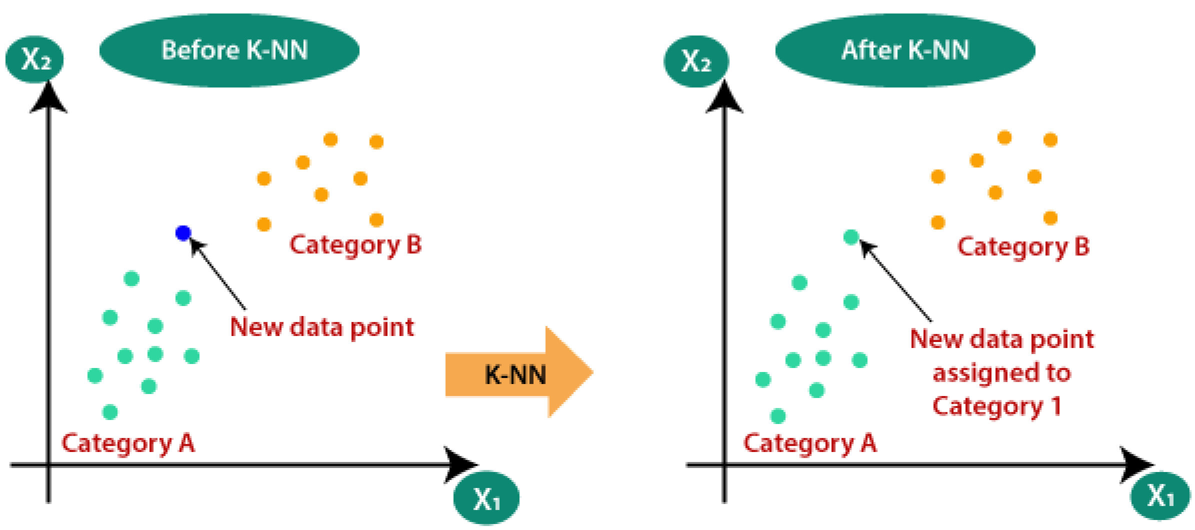


Illustration of how KNN makes classification about new sample according to (Liao, 2018).



The K-NN algorithm involves six steps:

Step 1: Choosing the neighbors' K-numbers is the first step.

Step 2: Determine the Euclidean separation between K neighbors.

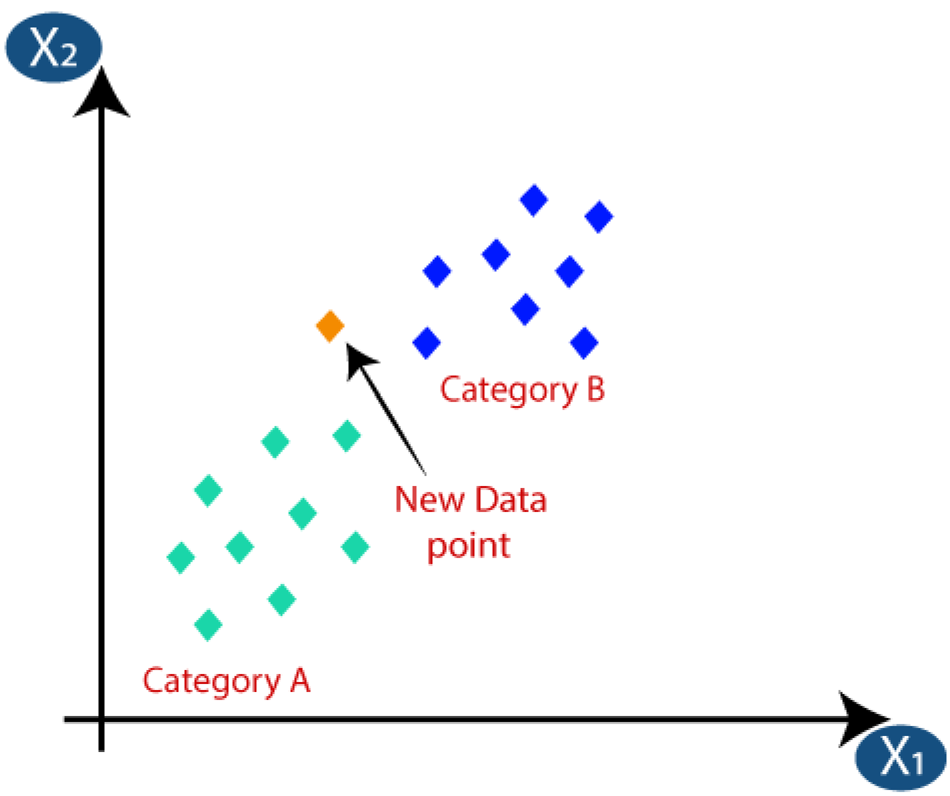
Step 3: Choose the K nearest neighbors based on the calculated Euclidean distance.

Step 4: Among these k neighbors, count the number of data points in each category.

Step 5. Assign the new data points to the category with the greatest number of nearby neighbors.

Step 6: The model is finished.

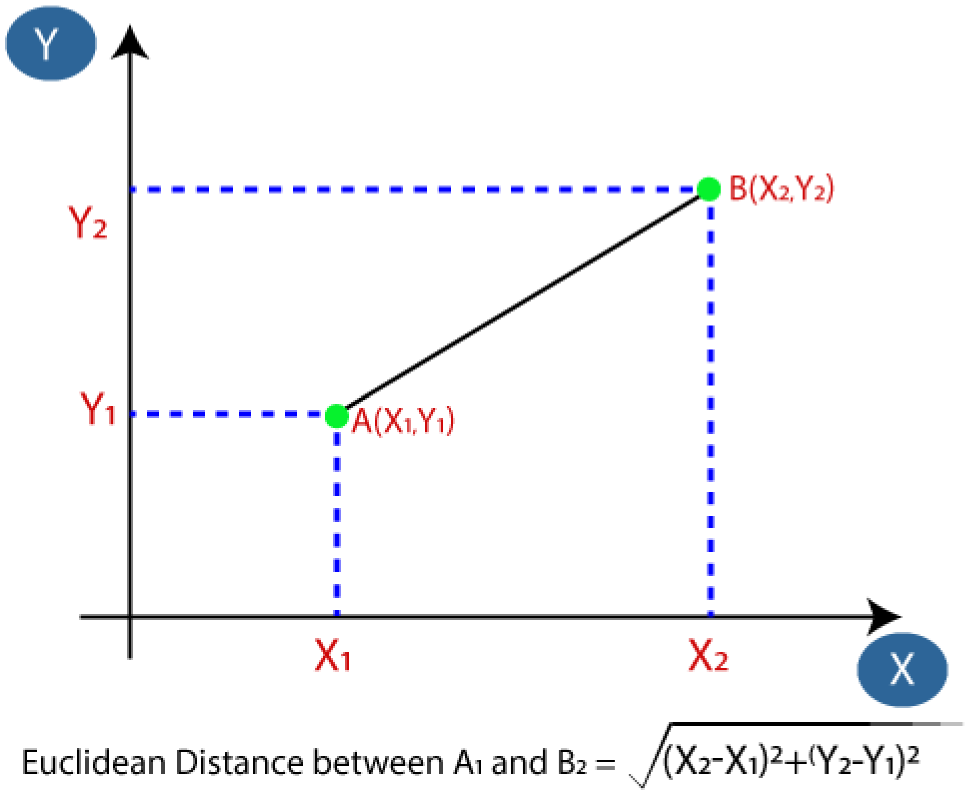
Example of KNN Algorithm Implementation



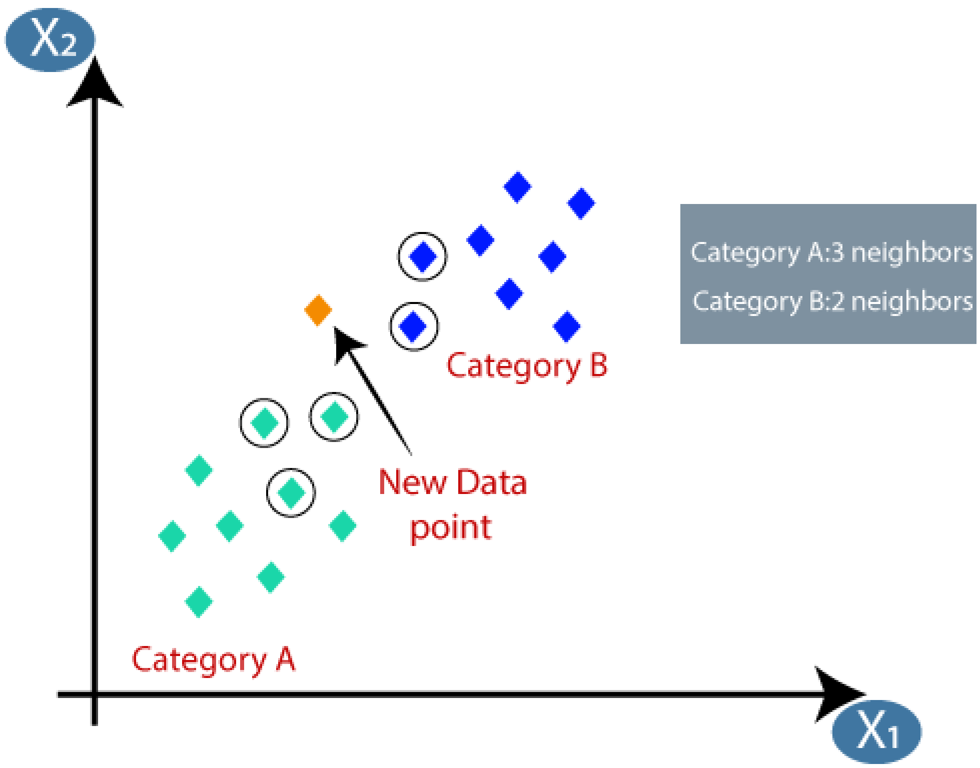
First, we'll decide on the number of neighbors; we'll go with k=5.

Then, we will calculate the Euclidean distance between the data points. The Euclidean distance, which we have already studied in geometry, is the separation between two points.

It is calculable as follows:



The Euclidean distance was calculated to identify the closest neighbors. According to the figure below, there were two closest neighbors in category B and three closest neighbors in category A.



This new data point must fall within category A because, its three closest neighbors are also from group A.

There are considerations when selecting the value of K in the K-NN algorithm. Since the ideal value for "K" cannot be established in a specified way, we must test out many possibilities to see which one performs the best. K is most accurately represented by the number 5. It's possible that a small number of K, like 1 or 2, will be noisy and lead to outlier effects in the model. K should have high values, but there might be some problems.

In the application of the algorithm in the base model, As soon as our features were prepared, this research adopted the Min-Max scaler to reduce the values. This helped reduce the bias for some books with too many features. The algorithm would find the median for all and equalize it.

```
from sklearn.preprocessing import MinMaxScaler
min_max_scaler = MinMaxScaler()
features = min_max_scaler.fit_transform(features)
```

After reducing the number of features, we can now construct our Python-based machine learning book recommendation system utilizing the K-Nearest Neighbor(KNN) algorithm method.

```
model = neighbors.NearestNeighbors(n_neighbors=6, algorithm='ball_tree')
model.fit(features)
dist, idlist = model.kneighbors(features)
```

We developed a machine learning model to recommend books, and now we must develop a Python function. The function would generate the similarity between one among others based on the weighted rating from the different categories.

```
def BookRecommender(book_name):
    book_list_name = []
    book_id = df2[df2['title'] == book_name].index
    book_id = book_id[0]
    for newid in idlist[book_id]:
        book_list_name.append(df2.loc[newid].title)
    return book_list_name

BookNames = BookRecommender('Harry Potter and the Half-Blood Prince (Harry Potter  #6)')
BookNames
```

3.4. Transfer Learning and customization to the system model

A machine learning agent's sampling efficiency may be considerably improved by using or transferring knowledge from previously learnt tasks for the learning of new tasks. A model created for one task is used in this research technique for a task that is linked to it.

In transfer learning, the learned features are first applied to a base network that was trained on a base dataset and task before being transferred to a second target network that is trained on a target dataset and task. The features should be general, i.e., applicable to both the base task and the target task, in order for this process to be successful.

Being that the base model captured the same parameters that were to be adopted in the custom model, the procedure of transfer learning was much simpler.

The process entailed classifying the fresh, unique dataset that was obtained from a respected institution's office of industrial attachment liaisons. The dataset received included information about the student's identity, the company they were affiliated with, and the company's rating. This stage involved classification and loading of the data due to the possibility of a corporation receiving numerous ratings.

```
company_id      title    authors  average_rating  ratings_count  num_pages  text_reviews_count   category  company_location  Country  language_code
```

Columns in the custom dataset included company_id, title, authors, average_rating, ratings_count, num_pages, text_reviews_count, category, company_location, country and language_code.

The fresh dataset was loaded after classification, and the base model was frozen. In order to account for the new columns in the custom dataset, the parameters were changed. The model was iteratively trained using the same algorithm, which was also utilized to produce results.

3.5. System Deployment

The recommender system was created using a hybrid ionic framework that combines online and mobile technologies. The systems' performance ranged from web browsers to android mobile devices thanks to the implementation of this framework. Python programming was used for the functional process flow and database's backend programming. The user interface and the system's result predictions are displayed on the front-end websites. The database uses primary keys to generate distinct user identifiers. The user data in the database is protected by key security encryption techniques. Checkers made of robots were used to confirm human behavior. This lessened the likelihood of gathering spam data.

3.5.1. Software to be used in the development

No	Software needed	Function of the software
1	M.S Windows 10	As an Operating System
2	Python and JavaScript Programming Language	Programming of the front interface and back-end functionality.
3	Google Chrome/ Mozilla firefox	As web client.

4	PyCharm	For interface design and python Programming.
5	Microsoft Word 2016	For project documentation.
6	Google Collaboratory	For running the models

3.5.2. Hardware to be used in the development

No	**Hardware needed**	**Purpose of the hardware**
1	Computer/ Laptop	Work station for developing the prototype.
2	Storage Devices (External Drive)	To store every project material, reports as a backup strategy.
3	Internet Connectivity	To upload the files to a remote server

The system contains Three major modules as admin module, Student module, Company module, each module has the same login page. The login page has a login username and password field. By entering values in that field, users should log in to the system.

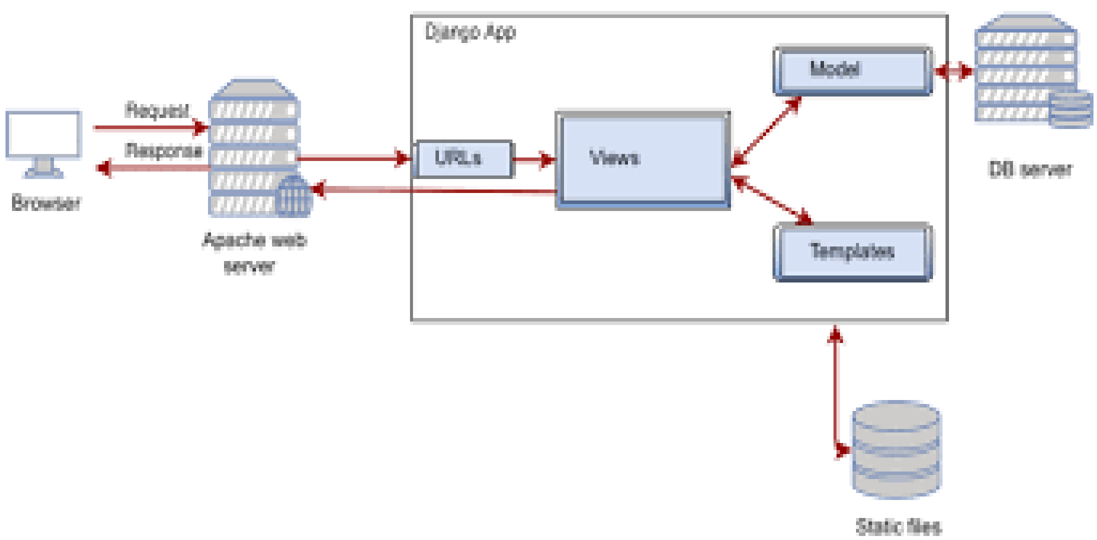


Figure 10: Overall App deployment System Architecture

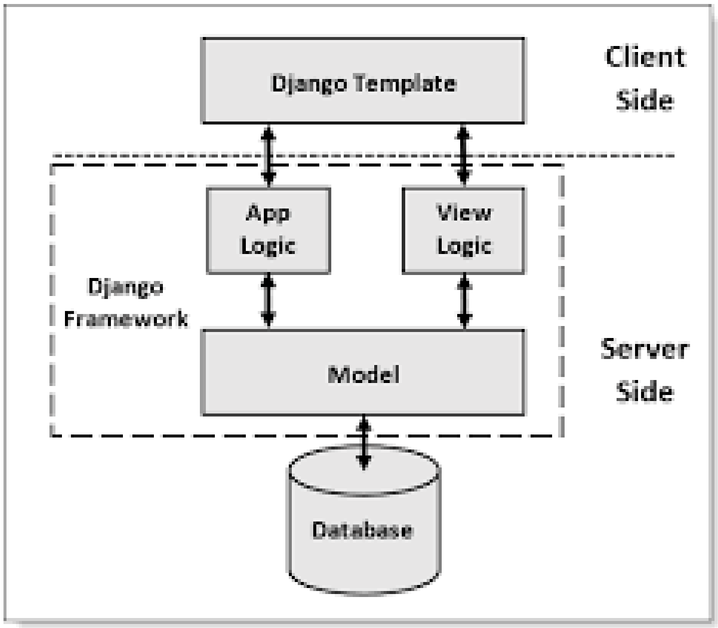


Figure 11: Client-side / Server-Side Architecture

Admin Module

The system is managed by the placement officer. The administrator is incredibly important to the system.

The following are things the admin can do:

I. Use a username and password to log in.
II. Continue the system-based placement operations.
III. Add businesses.
IV. Create or add new students.
V. Eliminate businesses or students.
VI. Suggest available students to various businesses.
VII. The administration can also evaluate each student's placement activity.

Student Module

The following activities are carried out by the students:

I. Students can create accounts on the student portal using their personal information.
II. Students may also upload their credentials for review by the business admin.
III. Revisit their profiles.

IV. Give feedback on the numerous businesses they visited in order to enhance customer experience.
V. If interested, students can apply to the company(s) of their choosing based on their preferences and credentials as listed in the advertisement.

Business module

The business administrator can do the following

I. Make advertisements for students who qualify.
II. Based on their qualifications - Determine whether a student is qualified for the opportunities that are offered.
III. Pick the right students.

Designing and implementing a web-based placement management system is one of this research's goals. High-quality placements are beneficial to students and have a positive effect on the colleges. The college finds it challenging and time-consuming to gather data from each student during this procedure. They typically gather data by hand. Working in the colleges' manual systems takes a lot of time and laborers. With the right login information, anyone inside or outside of universities, colleges, and other educational institutions can access the placement management system, an online application. The learner has the option to upload their data. The Model-View Template (MVT) pattern is used in the development of the web application in the Python Django framework. The system will feature many account kinds for various user types, including the principal, the head of department, placement officers and coordinators, and students. Each student receives a profile that includes the login information for the site. The system sorts the student's data in accordance with the requirements set forth by the various companies and manages databases using My SQLite. Additionally, a link will be sent via email to all qualified candidates so they may decide whether they want to attend the specific program.

The system's key benefit is that it may propose companies to students who want to apply to them. Companies can also obtain recommendations on students who are available whenever there are openings. However, the database's different parameters and preferences are used to provide recommendations. Age, gender, place of residence at the time, occupation, years of experience, and educational institution are some of these criteria and preferences. A machine learning tool that can forecast a student's chances of getting hired by various organizations based on the department's historical placement rates, his or her skill set, available grades, and other data.

Access Levels

Three modules make up the system: an admin module, a student module, and a company module. The login page is the same for each module. The system employs the CRUD method of programming. Create, Read, Update, and Delete are all implied by the acronym CRUD. Additionally, the system communicates backend and frontend capabilities using a model view template design. There is a username and password field on the login page. Users should log in to the system by entering values into that field. The student builds a profile by providing information about themselves, including their username, occupation, school, and current address. A genuine email address must be provided while registering because users are verified via their emails. The admin adds the company details into the systems and makes them available to the students. The company admins check the students profile and the uploaded curriculum vitae for eligibility to their company. Once the company has internship opportunities; the company admin submits the adverts to the system for selection by students. The students looking for internship can now react to the adverts if they are in line with their preferences. This ensures that no one is placed in the wrong area of practice or in an area not preferred by the student. The admin facilitates a smooth process of linking the students and the company. In addition, the students rate the companies based on their experiences during the internship programs. A student review on the available company is also necessary to ensure that companies give the best training programs to the selected students. A placement fee is also applied once a success placement is done via M-pesa. The student is notified via their email that he/she has been selected; the email contains the necessary requirements and information on how to embark on the internship system. On the students profile the student views their applications and their selection status. The system also notifies subscribed students any available adverts via their respective emails. Communication to all users is also made during the notification newsletter section. Users on the system can also reset and change password using validated emails that were used during sign up. Students and company’s admin receive messages once every task is completed on the website. Therefore, creating a user-friendly Graphical User Interface.

CHAPTER FOUR

RESEARCH FINDINGS AND DISCUSSION

4.1. Introduction

This chapter describes the implementation of the hybrid recommender system solution, called PlacementKe. The chapter is divided into sections that describe the hybrid models results, the system's design, modules, and components. The front-end and back-end components of the system are created through modeling the architectural design of the system. The initial PlacementKe prototype is made up of them. The creation of the graphical user interface, which includes the widgets, colors, and other obvious elements of the prototype, is the front-end of the hybrid model. The database design and multi-criteria algorithm implementation are examined in the back-end portion. We go into detail about the programming of the prototype later on in this chapter.

4.2. System Model Implementation

In the data cleaning of null values, it was observed that all of our scores fell between 0 and 5. Additionally, we learn more about the other columns, including their average mean scores and other details that may be useful to us moving forward. Additionally, we looked at the data types of each column and verified that our data did not contain any null values.

```
bookID                0
title                 0
authors               0
average_rating        0
isbn                  0
isbn13                0
language_code         0
  num_pages           0
ratings_count         0
text_reviews_count    0
publication_date      0
publisher             0
dtype: int64
```

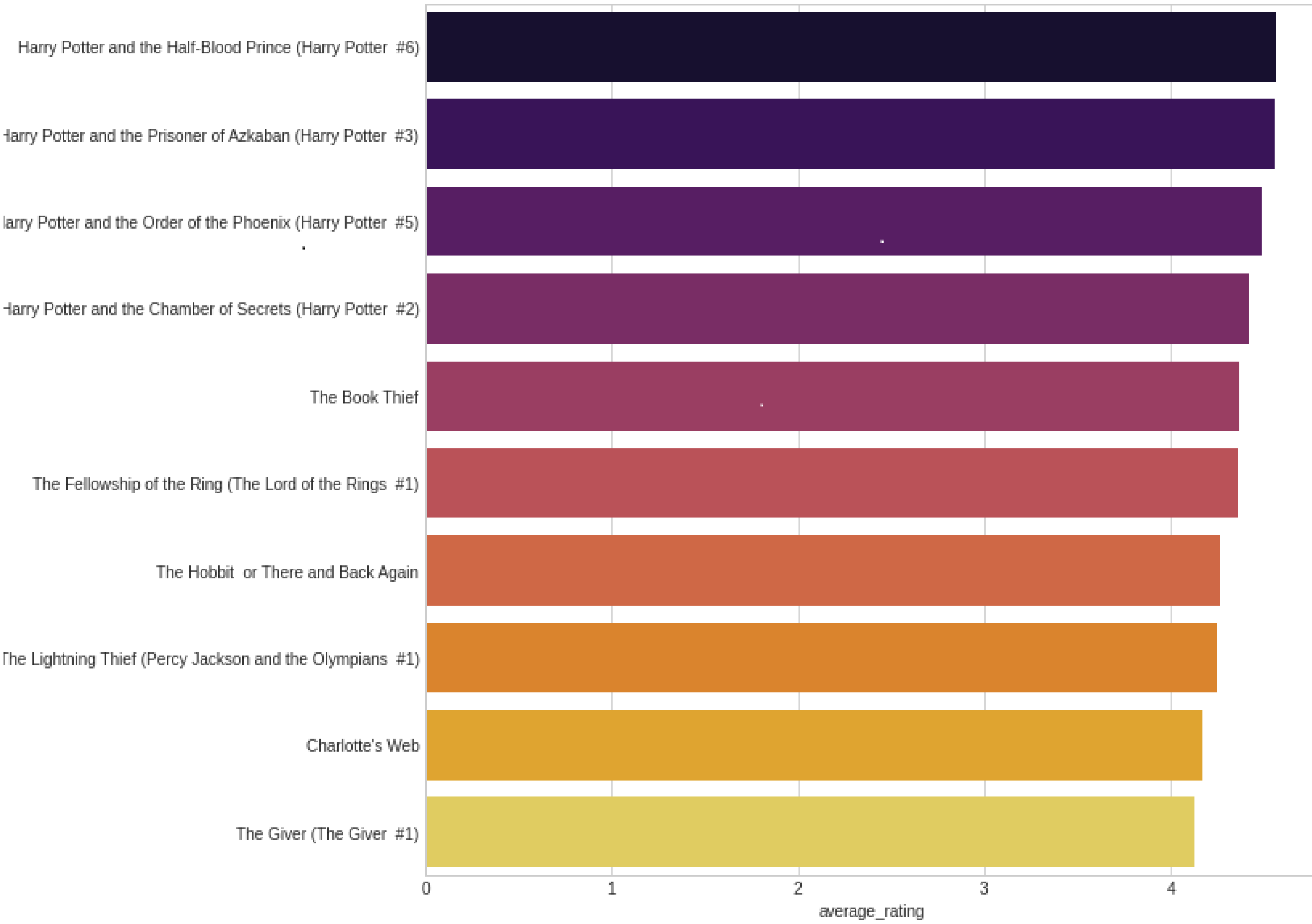
	bookID	average_rating	isbn13	num_pages	ratings_count	text_reviews_count
count	11123.000000	11123.000000	1.112300e+04	11123.000000	1.112300e+04	11123.000000
mean	21310.856963	3.934075	9.759880e+12	336.405556	1.794285e+04	542.048099
std	13094.727252	0.350485	4.429758e+11	241.152626	1.124992e+05	2576.619589
min	1.000000	0.000000	8.987060e+09	0.000000	0.000000e+00	0.000000
25%	10277.500000	3.770000	9.780345e+12	192.000000	1.040000e+02	9.000000
50%	20287.000000	3.960000	9.780582e+12	299.000000	7.450000e+02	47.000000
75%	32104.500000	4.140000	9.780872e+12	416.000000	5.000500e+03	238.000000
max	45641.000000	5.000000	9.790008e+12	6576.000000	4.597666e+06	94265.000000

Results from the classification of data according to the top ten books indicated that our data's maximum score was 5.0, but none of the books in the above result have a score that high. In fact, it was observed that the filtered books were based on how many notes they had. There may be some books in the data that are rated 5.0 but only have one or two remarks. Since the intention was to avoid certain publications in a perfect world, this type of filtering was employed.

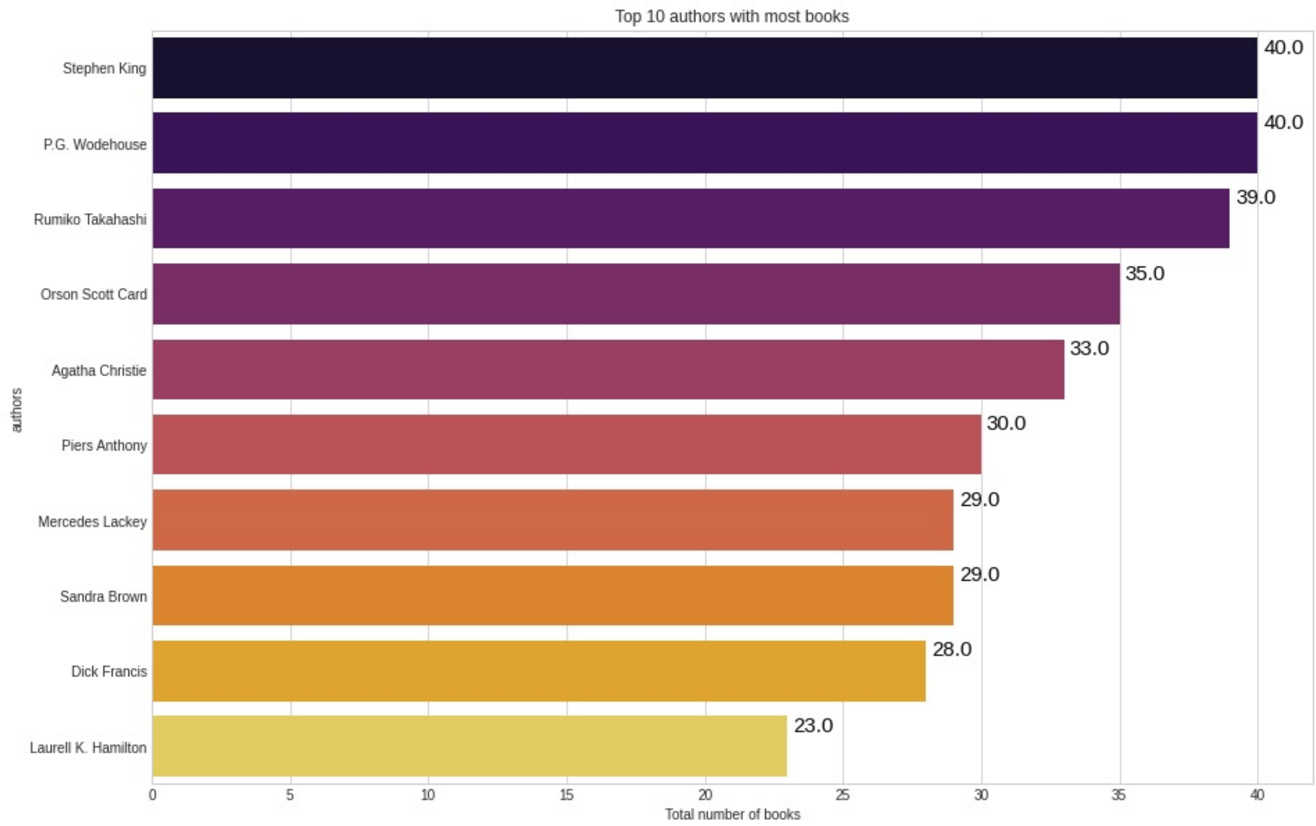
4.2.1. Data Visualization

Data was graphically represented in different categories in a bid to test the validity of parameters.

Top Authors based on quantity of publications



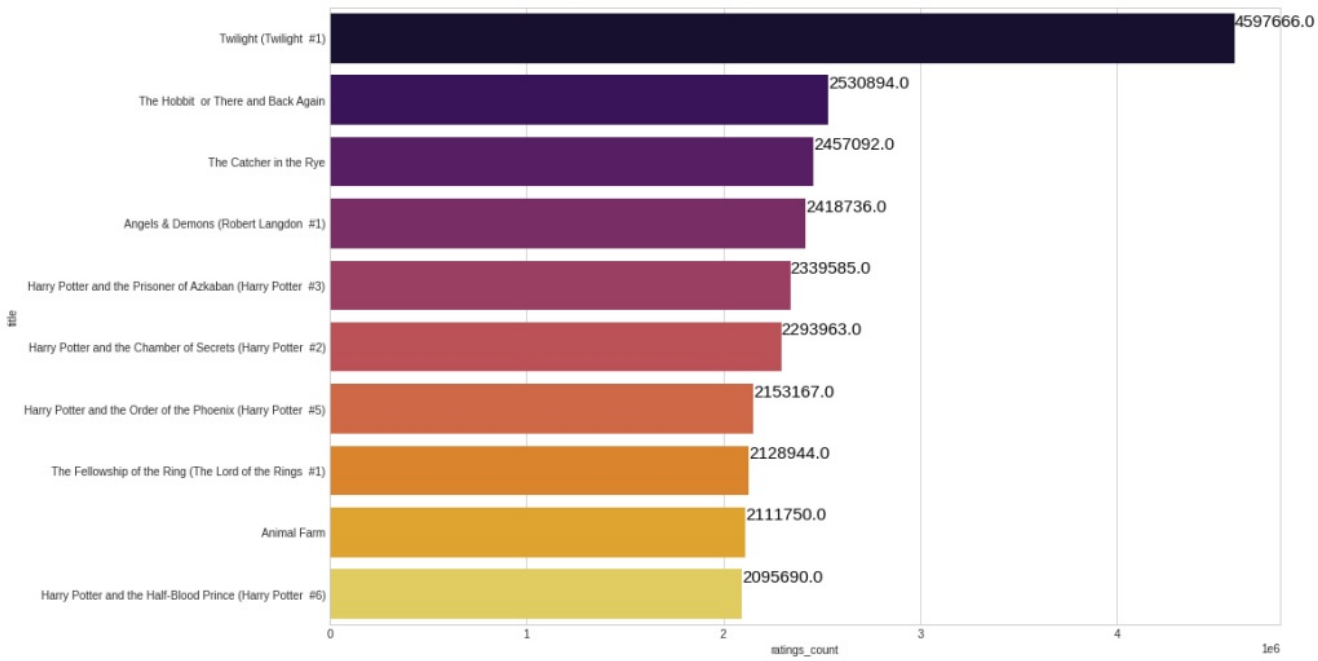
An examination of a few of the top authors in our data As long as the data for those books was available, they were ranked according to the number of books they had written.



According to the aforementioned chart, P.G. Wodehouse and Stephen King have the most books included. In our sample, both authors are followed by Rumiko Takahashi and Orson Scott Card, each with 40 novels.

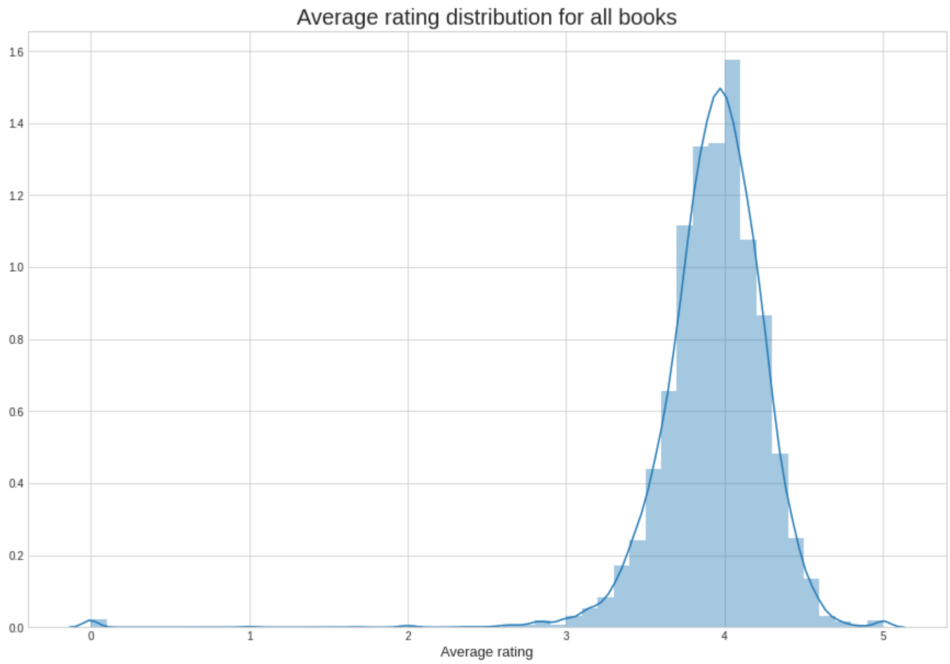
Top most reviewed books

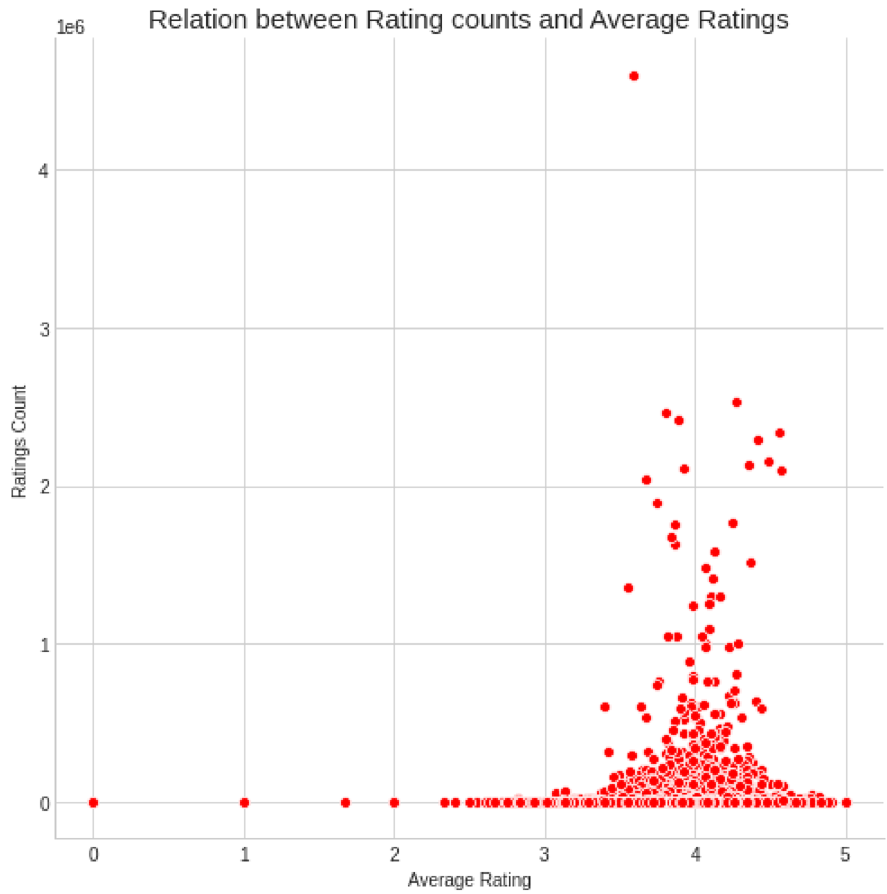
Our data includes both the average rating and the number of ratings for each individual book. N our study, this column was used to identify the books in our data that have received the most comments.

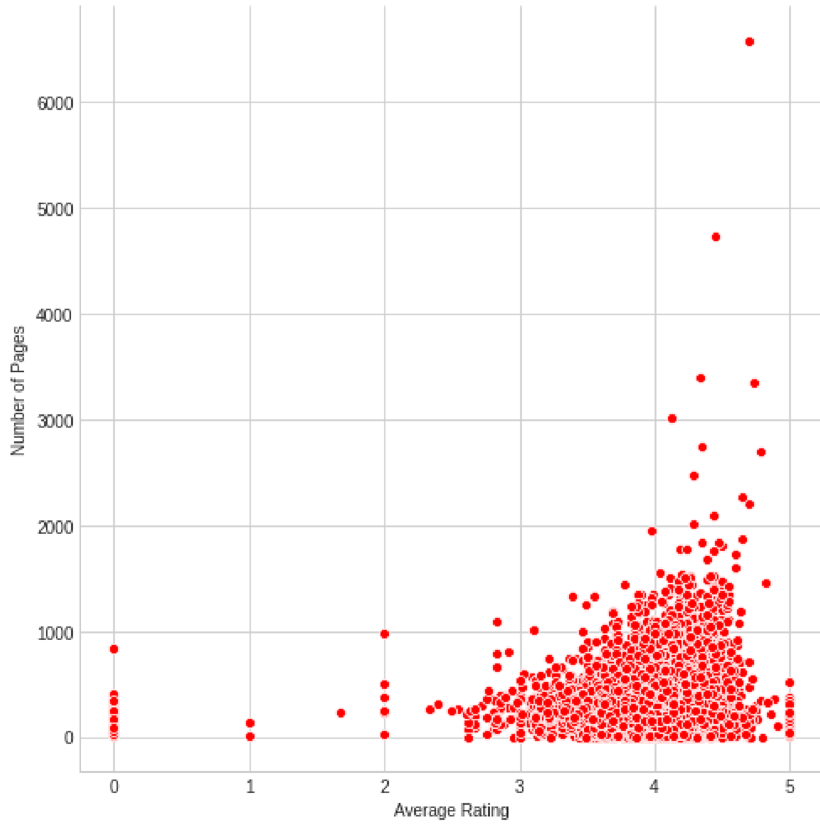


No other book has received as many ratings as Twilight! These ratings are all in the millions, as well! Accordingly, Twilight has received over 4 million reviews, followed by The Hobbit or There and Back Again and The Catcher in the Rye, which received over 2 million.

Average rating distribution







We can continue to use the language and number of ratings for our recommendation system after comparing the average rating with the various columns. We could omit the other columns because they didn't make much sense and wouldn't likely be of significant assistance to us.

4.2.2. Final Recommendation from weighted ratings.

After running the function that prompts the generation of recommendations, the system generated results for books similar to 'Harry Potter and the Half-Blood Prince (Harry Potter #6)' as illustrated below.

```
['Harry Potter and the Half-Blood Prince (Harry Potter  #6)',
 'Harry Potter and the Order of the Phoenix (Harry Potter  #5)',
 'The Fellowship of the Ring (The Lord of the Rings  #1)',
 'Harry Potter and the Chamber of Secrets (Harry Potter  #2)',
 'Harry Potter and the Prisoner of Azkaban (Harry Potter  #3)',
 'The Lightning Thief (Percy Jackson and the Olympians  #1)']
```

The above results indicate that the system not only weighted the ratings but also captured the book author. The top recommendations were accurate to the two parameters as the other recommendations relied on the average rating.

In a bid to validate the results, this study invoked a request for recommendations for books similar to 'Harry Potter and the Prisoner of Azkaban (Harry Potter #3)' and the results documented as shown below:

```
['Harry Potter and the Prisoner of Azkaban (Harry Potter  #3)',
 'Harry Potter and the Chamber of Secrets (Harry Potter  #2)',
 'Harry Potter and the Order of the Phoenix (Harry Potter  #5)',
 'Harry Potter and the Half-Blood Prince (Harry Potter  #6)',
 'The Fellowship of the Ring (The Lord of the Rings  #1)',
 'The Hobbit  or There and Back Again']
```

It was noted that the model produced outcomes that were comparable to those of the first request. This verified the model's capacity to consider ratings and identify the authors of the books. Given that this study will provide suggestions using weighted ratings as the secondary algorithm and user profiles as the anchor algorithm, it makes sense to incorporate the recommender algorithm into the custom model and evaluate the outcomes.

4.2.3. Evaluation of the custom model

The custom model was implemented using a custom dataset obtained from a Industrial Attachment Liaisons Office in a reputable institution. The Data contain sampled names of students and ratings of different companies based on 5 major categories of industries such as technology, education, audit, banking and architecture.

It was observed that no null values were captured since the scores fell between 0 and 5. Additionally, we looked at the data types for each column and confirmed that our data did not contain any null values.

```
company_id            0
title                 0
authors               0
average_rating        0
ratings_count         0
num_pages             0
text_reviews_count    0
category              0
company_location      0
Country               0
language_code         0
dtype: int64
```

	company_id	average_rating	ratings_count	num_pages	text_reviews_count
count	30.000000	30.000000	3.000000e+01	30.000000	30.000000
mean	15.500000	4.214333	3.345079e+05	652.733333	5118.000000
std	8.803408	0.346377	7.406214e+05	755.688219	9535.959392
min	1.000000	3.440000	1.900000e+01	6.000000	1.000000
25%	8.250000	3.902500	2.285250e+03	231.500000	146.000000
50%	15.500000	4.290000	2.836550e+04	343.500000	653.500000
75%	22.750000	4.500000	9.607700e+04	815.000000	3885.250000
max	30.000000	4.780000	2.339585e+06	3342.000000	36325.000000

Since majority of the parameters did not change, this study proceeded to implement the model and test the results. A sample test involved generating recommendations for companies with similarity to 'Airtel Kenya' based on the weighted ratings and category. The results were as illustrated below:

Top three recommendations related to Airtel Kenya.

```
'Airtel Kenya',
'Safaricom Limited',
'Techobrain Group',
```

To validate the results, we generated results related to 'Safaricom Limited' and illustrated as show below:

Top three recommendations related to Safaricom Limited.

```
'Safaricom Limited',
'Airtel Kenya',
'Techobrain Group',
```

These results allow us to conclude that the weighted average of company ratings and category are both perfectly effective in producing similarity indices. The user will be able to provide recommendations based on their profiled field of study and top-rated businesses after it has been included in the prototype.

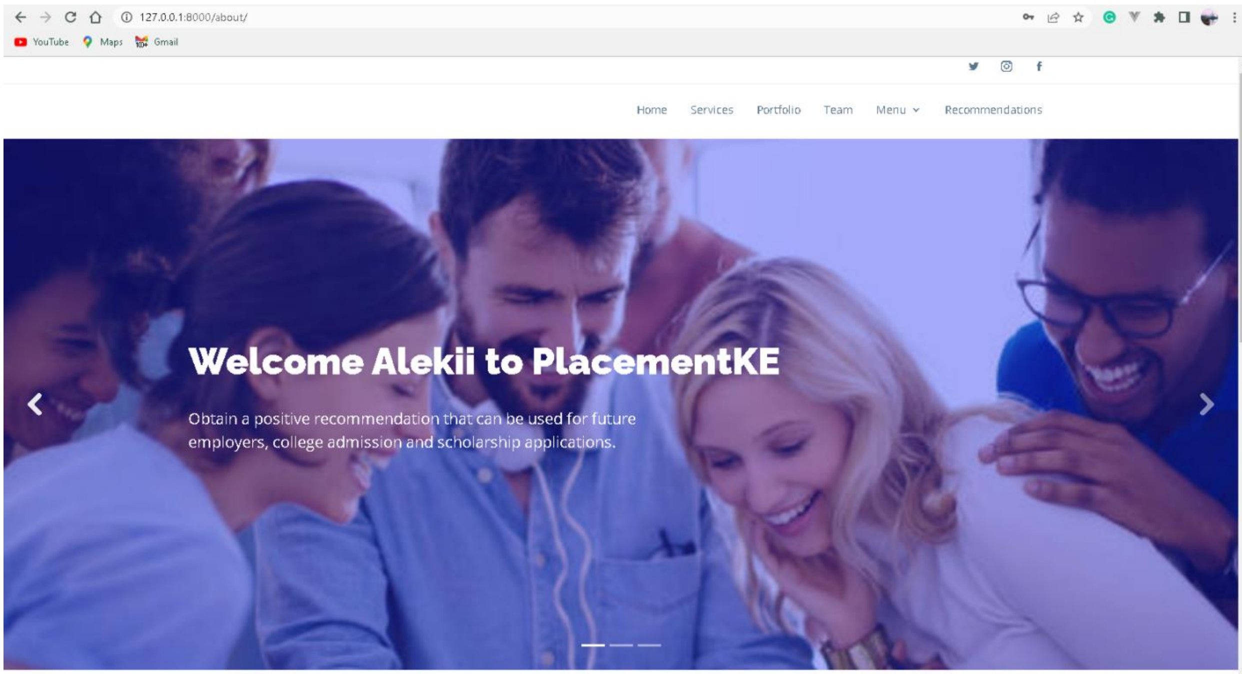
4.3. Model Deployment

Python code was used to implement the custom model code on the prototype. To produce predictions at the background level, the model was tunneled. To give consumers an interactive platform, the graphical user interface was created using JavaScript.

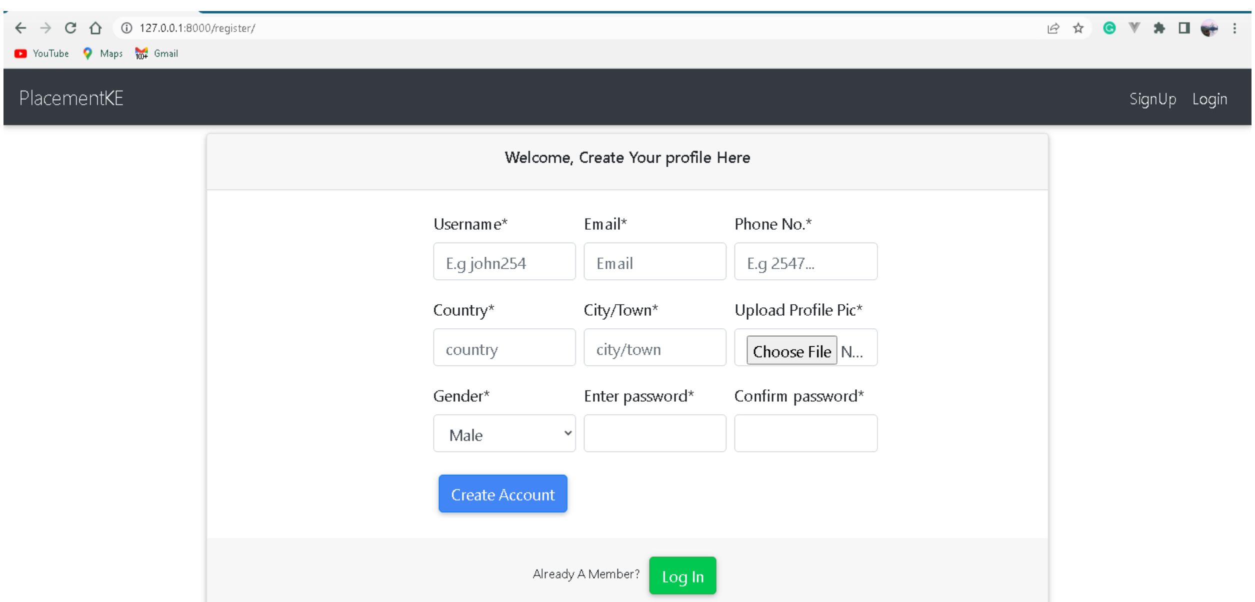
4.3.1. System View

The landing Section with the main navigation bar

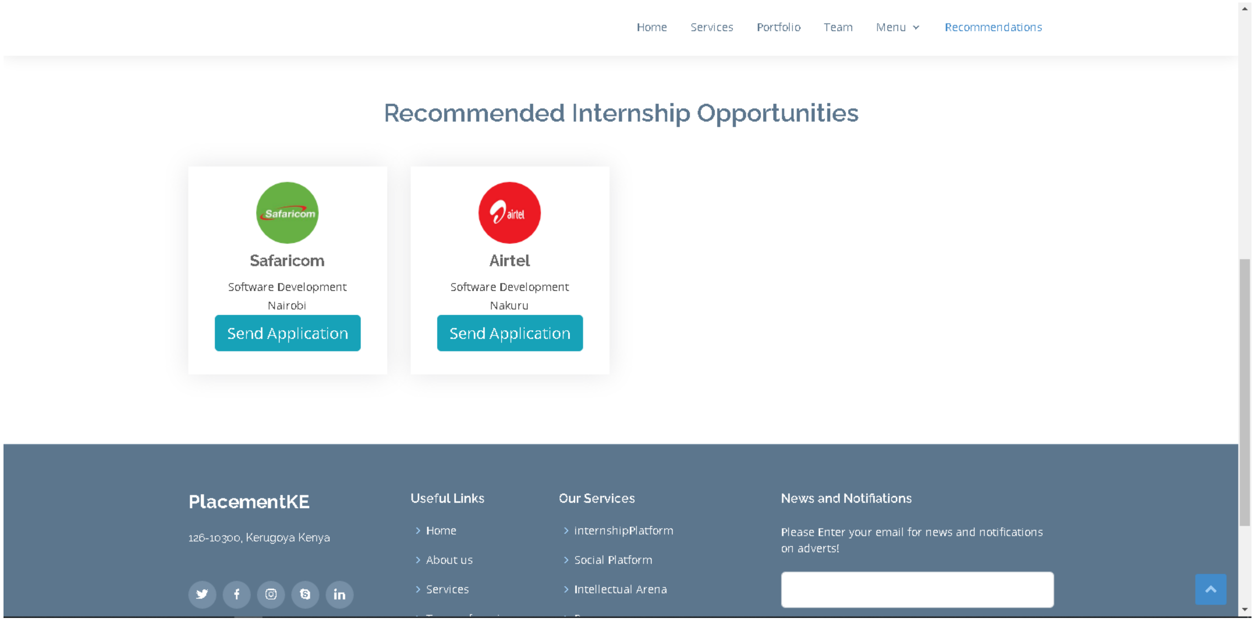
Landing Page



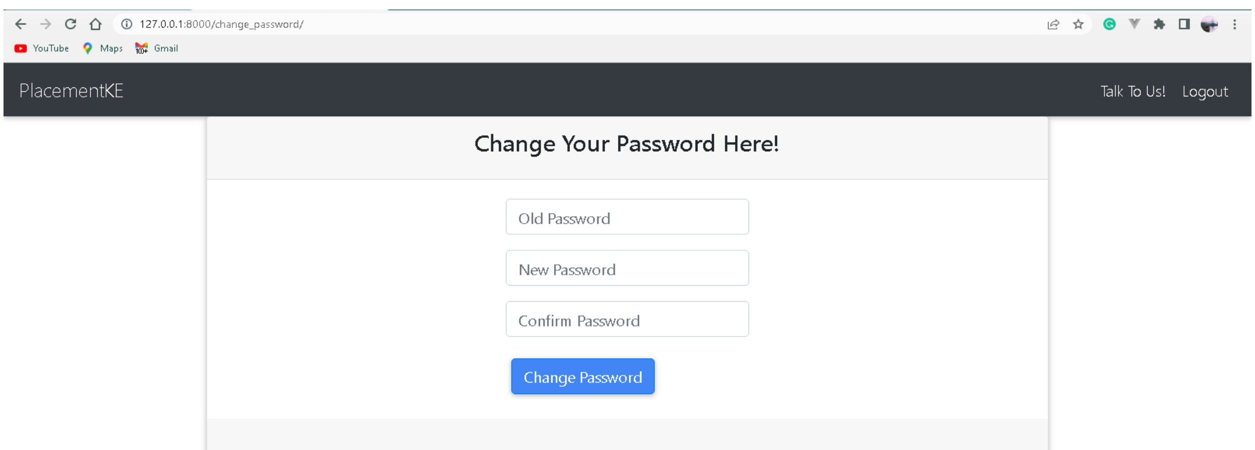
Registration Section



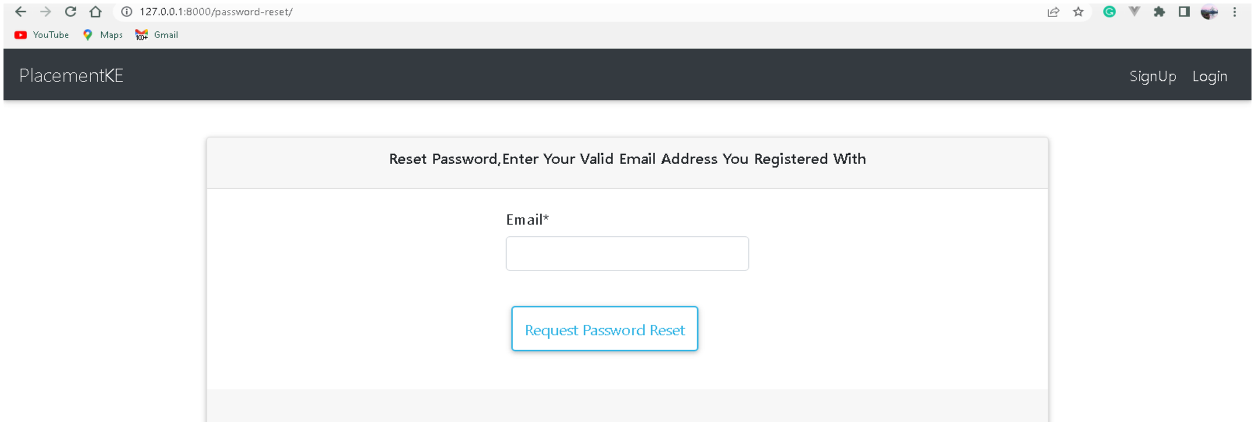
The Recommendations Section



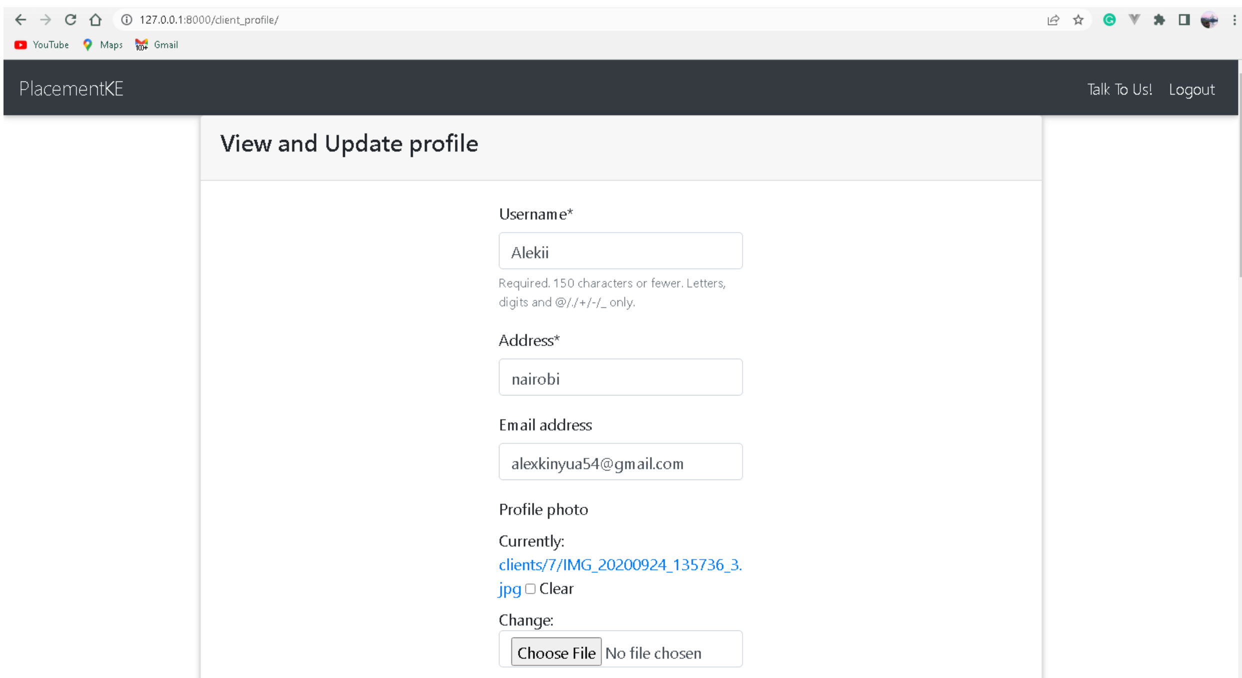
Password Change Section



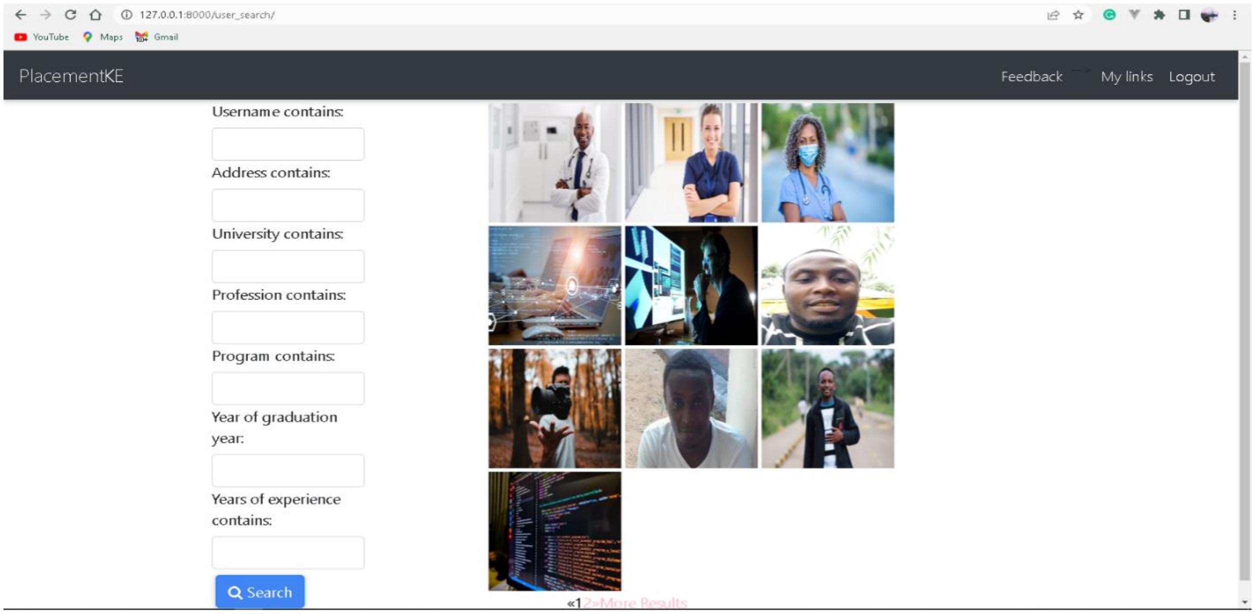
Reset Password Section



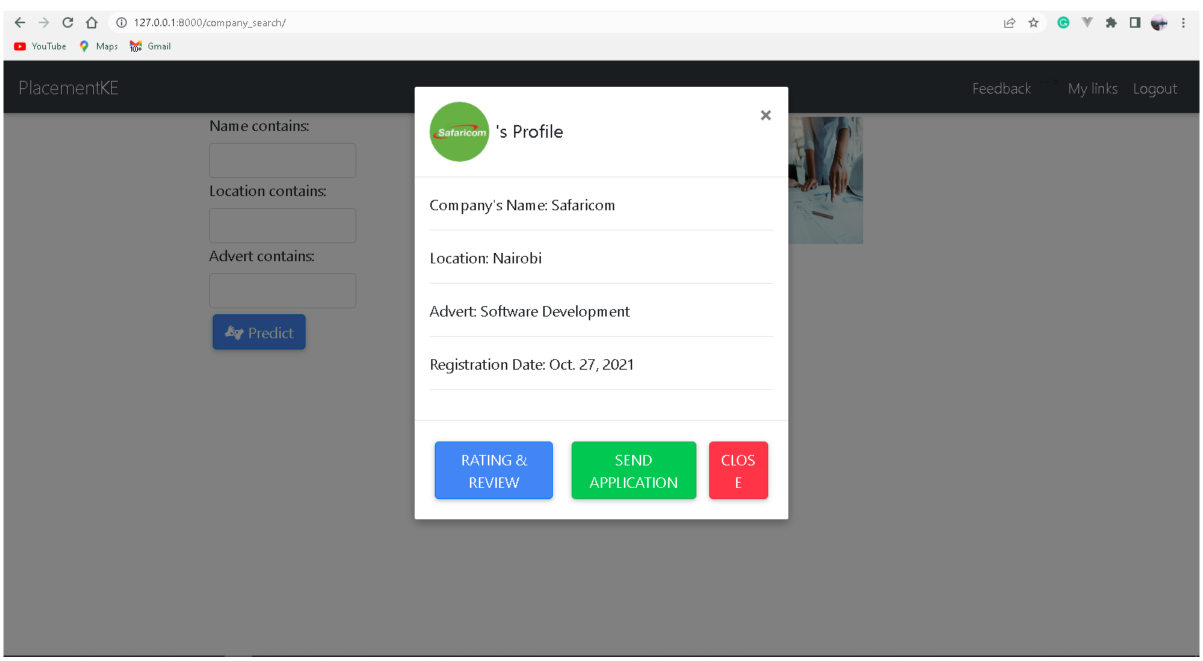
Student's View and update Profile



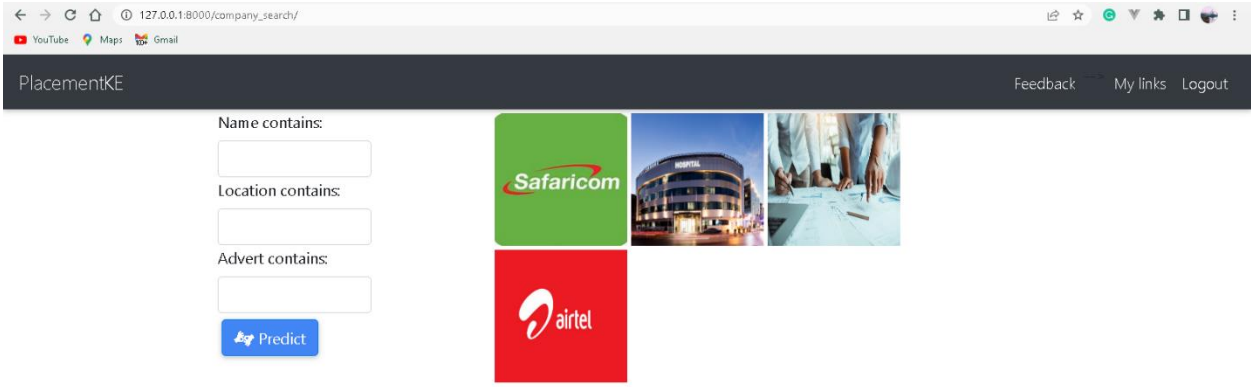
Students Selection and Filtering for Recommendations



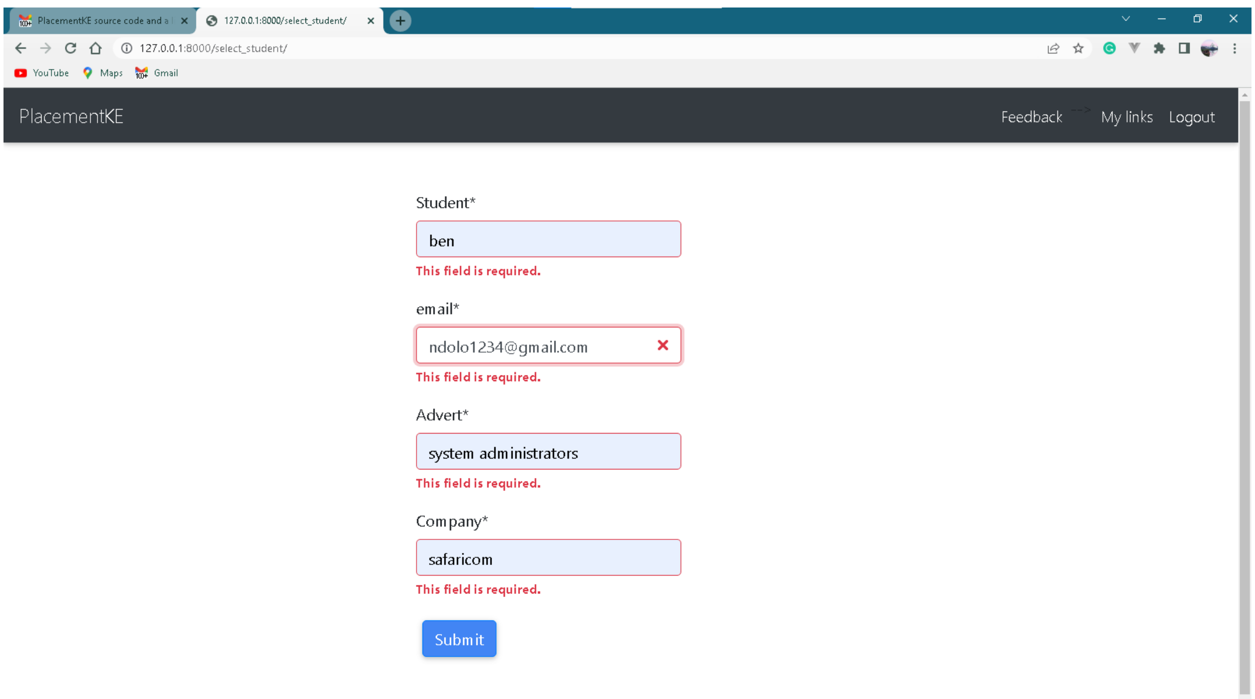
Company's Profile



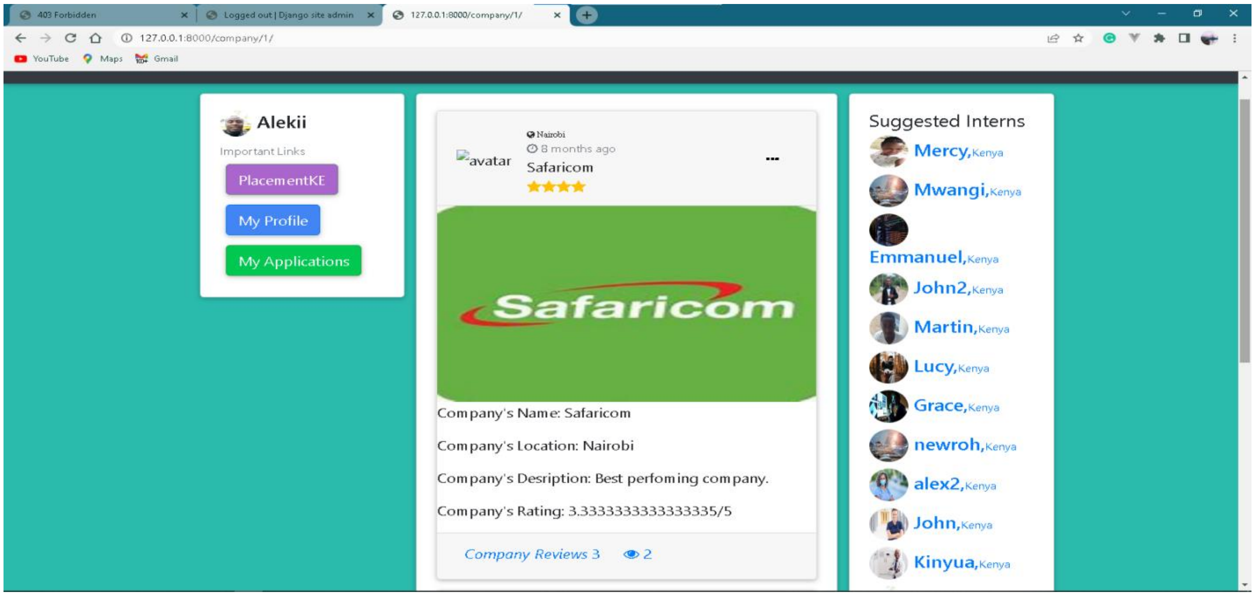
Company Selection and Filtering

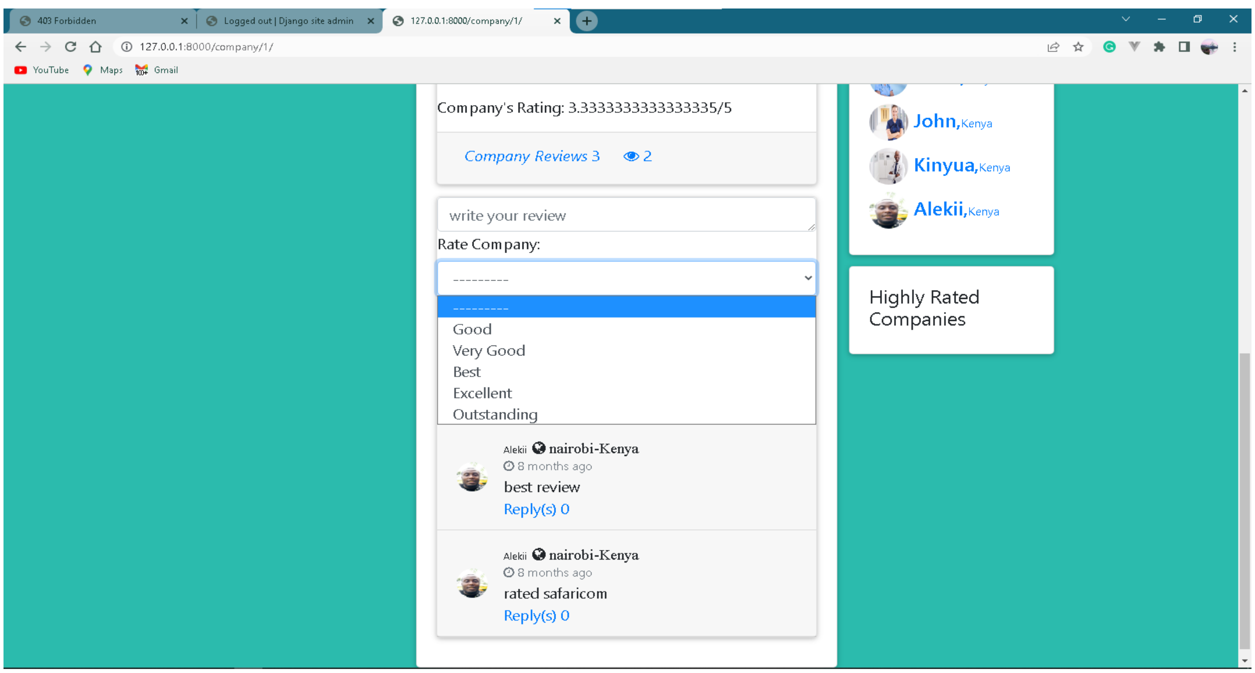


Student's Selection page

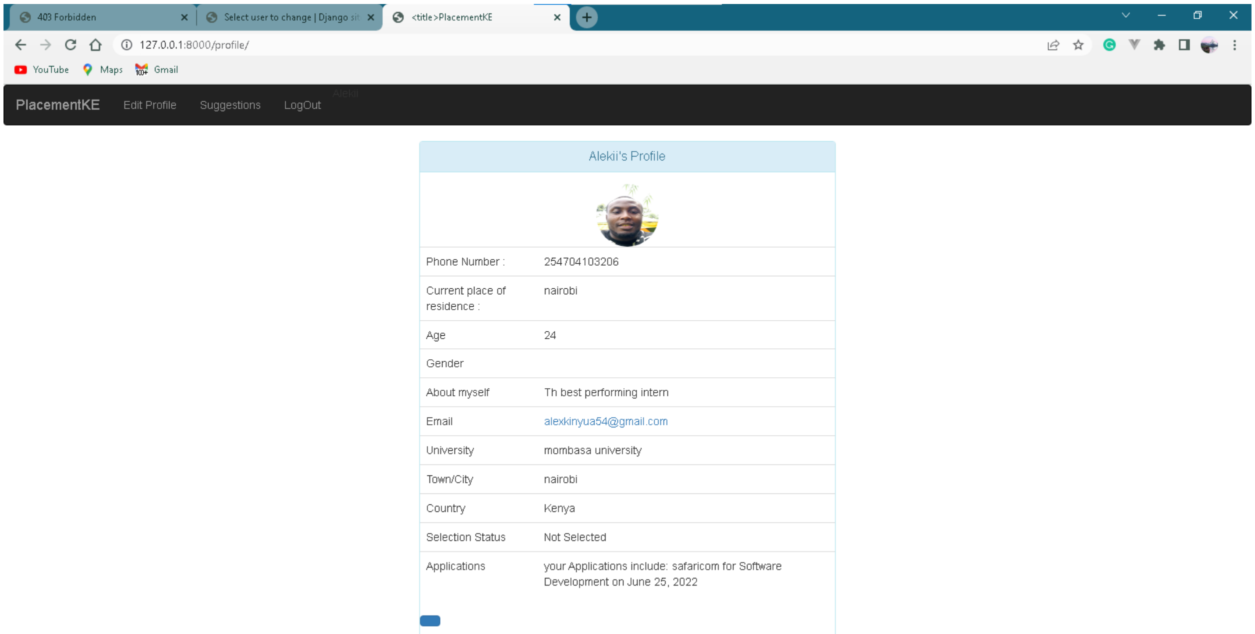


Company Reviews, Ratings and Suggested Interns

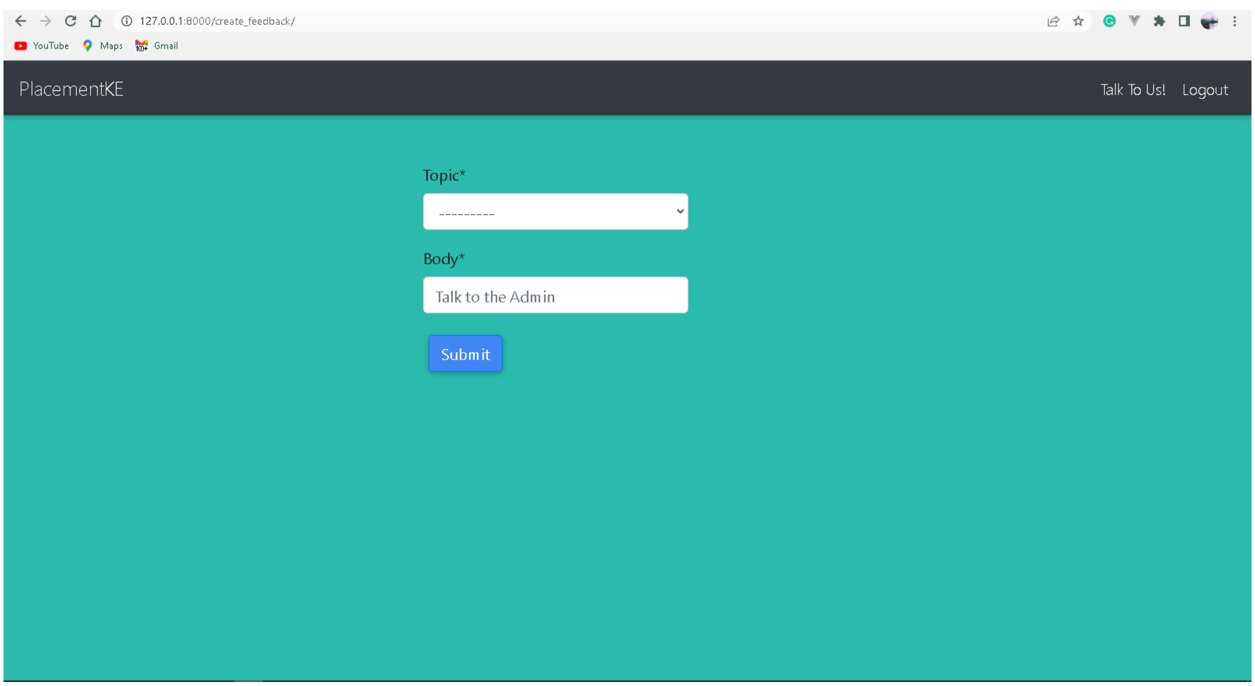




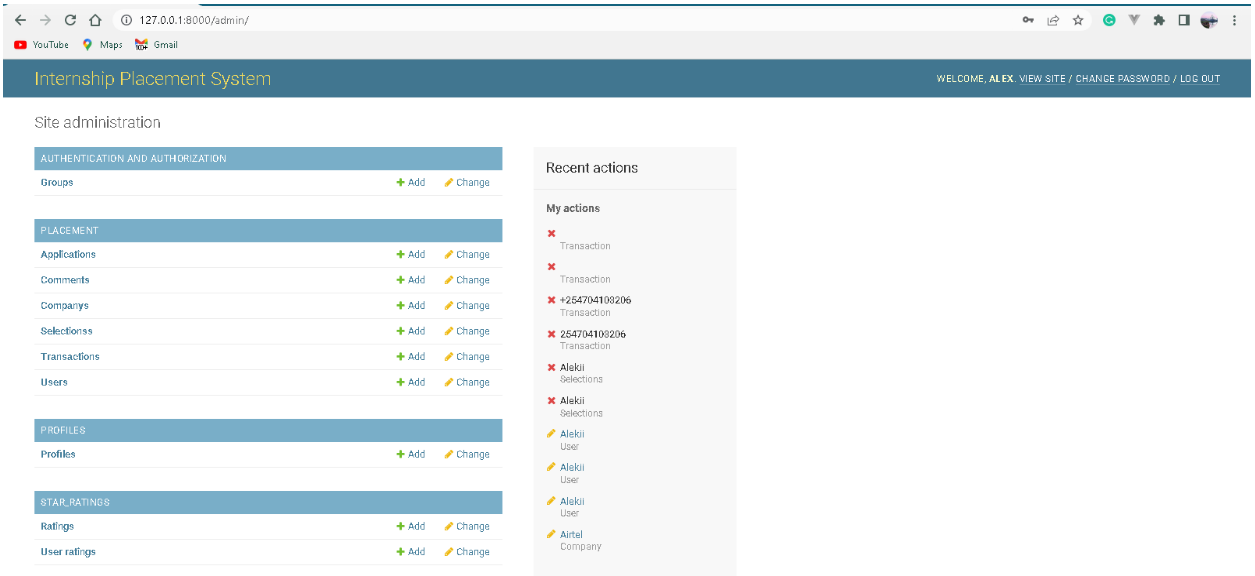
Student's Placement Status and Applications Sections



Admin Feedback Section.



Admin Panel



4.4. Feedback Analysis

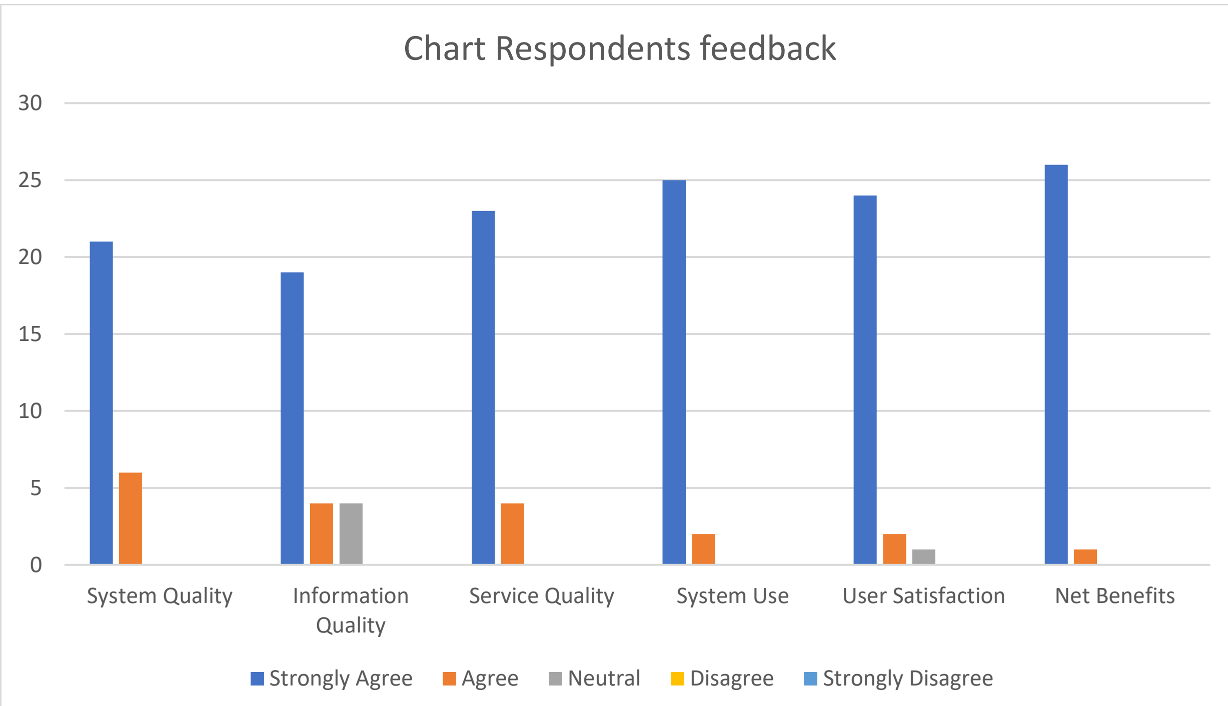
A total number of thirty students were sampled from the Machakos University Industrial Liaisons Office to test the system experience. The sampled respondents were placed in 3 different categories targeting those who have already undergone for industrial attachment, those who are presently in industrial attachment and those who are yet to undertake industrial attachment.

The outcome was tabulated as illustrated below.

Description	***Target Number***	***Total responses (%)***
Those who have undergone industrial attachment	10 students	33.33%
Those presently undertaking industrial attachment	9 students	30%
Those yet to undertake active industrial attachment	8 students	26.67%
Those who never responded	3 students	10%

The detailed summary of responses was tabulated showing respondents feedback in each category: -

Core Parameters	*Strongly Agree*	*Agree*	*Neutral*	*Disagree*	*Strongly Disagree*
System Quality	21	6	0	0	0
Information Quality	19	4	4	0	0
Service Quality	23	4	0	0	0
System Use	25	2	0	0	0
User Satisfaction	24	2	1	0	0
Net Benefits	26	1	0	0	0



In the first question, 77.78% of the respondents strongly agreed that the system quality is up to standard. 22.22% of the respondents agreed with no other respondent giving a dissenting view.

In the second question, 70.37% of the respondents strongly agreed to the quality of information found in the prototype. Only 14.81% of the respondents agreed and a similar number who remained neutral. In the third question, 85.19% of the respondents strongly agreed to the quality of service rendered via the prototype. 14.81% of the respondents just agreed with this view. We observed no other opinions raised. In the fourth question on system use, 92.59% strongly agreed on the ease of use and navigation within the system. Another 0.07% responses supported this view with no one

dissenting it. The fifth question captured the level of user satisfaction with 88.89% of the respondent's showing satisfaction in the system. 0.07% also agreed and another 0.03% remaining neutral. In the survey on the overall benefit of the system, 96.30% of all the respondents strongly agreed that the system will have a huge impact to both the individual and organization.
Following the Delone and McLeans model of evaluating systems, three factors are dependent on others. In this case the independent variables of system quality, information quality and service quality contribute to the dependent variables of user satisfaction ad systems use. These two dependent variables determine the overall net benefits to the individuals and organizations.

Based on our analysis above, the first 3 parameters give a positive average of 77.78%. This average shows that respondents appreciated the systems use which further contributed to an overwhelming user satisfaction. Feedback from our respondents showed the value of the prototype and impact it would generate on a comparison level with the existing systems and methodologies of industrial attachment placements.

4.5. Limitations and Challenges

One was that there wasn't much time for development. We haven't been able to put all of the prototype's modules into practice due to the short University time limitations for research projects. Second, the MPESA Payment Module installation was unsuccessful due to bureaucratic systems. The installation of the payment sections was hampered by the cumbersome procedures for obtaining Payment APIs.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1. Recommendations for further improvements

This research identifies two recommendations for further improvements of the solution output. One is the Implementation of payment methods for users. This research recommends the implementation of payment modules to facilitate the posterity of the system implementation especially when large data storage is needed.

Second is the Integration of an AI bot that can search through the web and collect data by notifying the opportunity publisher to send or advertise using the platform.

5.2. Conclusion

In an effort to enhance the industrial placement experience, this study suggested using recommender systems. The results of the literature review described the lengthy process that students had to go through in order to get access to possibilities for industrial attachment. A significant situation where a student pursuing academic credentials was unable to acquire an appropriate industry attachment training opportunity stood out as a notable feature. The bulky nature of the procedure was highlighted in the research as a gap that would be filled by starting this project. In accordance with the study's major goal, recommender systems were found to be the best way to enhance the user experience in the age and era of digital transformation.

Due to the success of currently operating recommendation systems like those used by Netflix, LinkedIn Inc., and Indeed Inc. This piece pleaded for an architecture that would view the virtual experience provided as a placement service. It was noted that an algorithm filtering technique would subsequently be used to give the placement service. The three main categories of filtering strategies, including content-based filtering, collaborative filtering, and knowledge-based hybrid systems, were observed. Due to its success in recommendation engines across several industries, including e-recruitment and e-commerce, our research adopted a hybrid collaborative filtering.

The design science approach was chosen as the model employed in the technique in an effort to implores it. According to the study, the design science approach would be an end-to-end model that would not only allow for a description of the problem statement but would also cast a close eye on improving the system artifact. The research strategy was determined to have a clear progression of tasks from the problem description to the finished product. The quick software development approach was used to create the prototype in an effort to match the research design. In the literature review, the software development approach was noted as having been successful in prototype development.

It was reported that the approach consists of six parts, including problem analysis, objective identification, prototype construction, testing, and evaluation. While the remaining phases were categorized during the prototype stage, the first two steps were decided during the study during the literature review. It was also highlighted that the implementation structure for the prototype's development included assessing user requirements, designing a system prototype, and iteratively testing while modifying the intended goals.

The prototype's implementation was built in three stages based on the research findings. The first of the three processes involved creating a base model that would be trained and used to provide the right outcomes. Using secondary data, the custom model was then trained using the underlying model to make sure it functions as expected. For api tunneling to the graphical user interface, the custom model was then loaded in Python. It was found that both the base model and the modified model produced comparable outcomes. This leads one to the conclusion that the research's development goals were accomplished.

The model successfully mastered the algorithmic structure and produced a high index rate percentage when evaluated on sample data from Kaggle and Machakos University Industrial Liaisons Office. The model was evaluated with actual data after deployment, and the results were accurate representations of what the model had output during the early phases of implementation. In an effort to achieve the main goal, the system was assessed by a group of sampled users, and more than 70% of them found it to be a solution for enhancing the student industrial placement process in the current era of online learning. When evaluated on actual data, the constructed prototype produced results that were pertinent to the problem statement, with each user producing recommendations that were pertinent.

The built prototype generated relevant findings with each user producing pertinent recommendations. This leads one to the conclusion that the research's development goals were accomplished. This study finds that, independent of their operational scope, multi-criteria hybrid collaborative filtering recommender systems are capable of improving the student industrial attachment placement process.

References

Attarde, D. V., & Prof. Manmohan, S. (2017, December). Survey on Recommendation System using Data Mining and Clustering Techniques. *International Journal for Research in Engineering Application & Management (IJREAM), 03*(09).

Awani, N. (2018, March). *Companies offering attachments in Kenya in 2018*. Retrieved from Tuko.co.ke: https://www.tuko.co.ke/266408-companies-offering-attachments-kenya-2018.html#266408

BrighterMonday. (2018, 10 10). *Industrial Attachment/Internship Opportunities.* Retrieved from BrighterMonday Kenya: https://www.brightermonday.co.ke/job/industrialattachmentinternship-opportunities-4845jv

Capital Campus. (2014, September 3rd). *Students: 5 things you need to do to secure an attachment.* Retrieved from Capital FM: https://www.capitalfm.co.ke/campus/students-5things-you-need-to-do-to-secure-an-attachment/

Kamaul, P. W., Wamutitu, J. M., & Mbugua., G. (2013). Employability of Fashion and Garment Making Students from 22. *Vetri Education (Quarterly Journal in Education), 8*(4), pp. 22-30.

Kiplagat, H., Khamasi, J. W., & Karei, R. (2016, August). Students' Experience of Industrial Attachment: A Case of a Public University. *Journal of African Studies in Educational Management and Leadership, Volume 7*(Sn 1), 82-97.

KMTC. (2018). *About KMTC*. Retrieved from Kenya Medcial Training College: https://kmtc.ac.ke/site/about-kmtc/

KNBS. (2013). *University Enrollment.* Retrieved from Kenya National Bureau of Statistics: https://www.knbs.or.ke/download/university-enrolment/

LinkedIN. (2019, July 24th). *About LinkedIN*. Retrieved from LinkedIN: https://linkedin.com

Mc'Odongo, L. (2019, July 2nd). *Attachment / Internship Hustle*. Retrieved from Linking Industry With Academia(LIWA): https://www.liwaprogrammetrust.org/blog/3-Attachment-Internship-Hustle.html

Moore, L. S. (2017). e-Recruitment: Creating an Effective Job Recommender System. *The University of the West Indies*.

Muchemi, a. k., Muthoni, d. m., Mutahi, i. w., Gunga, s. o., & Origa, j. o. (2013). The Implications of Collaborative Industrial Attachment for Kenya Vision 2030 Development Programmes. *African Journal of Education and Technology, 3*(1), 57-67.

Muthoni, D. M., Gunga, P. S., Mutahi, I. W., & Origa, D. J. (2016). Influence of Industrial Attachment on the Competence of Instructors and Students in Enabling Creative Innovations for Industrialisation in Kenya.

NITA. (2018). *Who we are*. Retrieved from National Industrial Training Authority: https://www.nita.go.ke/about-us/who-we-are.html

Nyamwange, J. (2016). Influence of Student's Interest on Career Choice among First Year University Students in Public and Private Universities in Kisii County, Kenya. *Journal of Education and Practice, 7*(4).

Omankwu, O. C., Nwagu, C. K., & Inyiama, H. (2018, April). Computer Science Research Methodologies. *International Journal of Computer Science and Information Security (IJCSIS), 16*(4).

Rahul , H., Kulvinder, S., & Sanjeev, D. (2014, July). A Study of Recommender Systems on Social Networks and Content-based Web Systems. *International Journal of Computer Applications, Volume 97 - No. 4*, 0975 – 8887.

Shaha , T. A.-O., & Mourad, Y. (2012, July 26th). A survey of job recommender systems. *International Journal of the Physical Sciences, Vol. 7(29)*, 5127-5142.

Shahab, S. S., Jamshed, S., & Rashid, A. (2017, September 18). Classifications of Recommender Systems: A review. *Journal of Engineering Science and Technology Review, 10*(4), 132 - 153.

Tanya, M., Shikha, G., & Dr. Atul, M. (2018, January). A Survey on Recommendation System. *International Journal of Advance Research in Science and Engineering, Volume No. (07)*(Issue No. (03)), 439 - 444.

Teng, W.-G., & Lee, H.-H. (2007, January). Collaborative Recommendation with Multi-Criteria Ratings. *Journal of Computers, Vol.17, No.4,*.

Wenxing, H., Siting, Z., & Huan, W. (2013, August). A Job Recommender System Based on User Clustering. *JOURNAL OF COMPUTERS, 8*.

Zheng , S., Hong, W., Zhang , N., & Yang , F. (2012). Job recommender systems: A survey. *7th International Conference on Computer Science & Education (ICCSE)*, 920-924.

Appendix 1: Database structure and code

```
class Company(models.Model):

ADVERT_CHOICES = (

        ('Doctor', 'Doctor'),

        ('Nurse','Nurse'),

        ('Teacher','Teacher'),

        ('Civil Engineer','Civil Engineer'),

        ('Software Development','Software Development'),

    )

    #user = models.ForeignKey(User,related_name='user_feedback',on_delete=models.CASCADE)

    name = models.CharField(max_length=30)#,choices=COMMENT_CHOICES)

    location = models.CharField(max_length=300)

    user_admin = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)

    advert = models.CharField(max_length=300,choices = ADVERT_CHOICES, blank= False )

    email = models.CharField(max_length=30)

    description = models.TextField(max_length=400,  blank = False)

    registration_date = models.DateField(default=timezone.now)

    company_rating = models.PositiveIntegerField(default=0, validators=[MinValueValidator(1),
MaxValueValidator(5)])

    company_review = models.CharField(max_length=300)

    profile_photo = models.ImageField(upload_to=cover_upload_path, blank=True,
default='clients/None/IMG_20190831_123316_G2XdrKx.jpg ')


class User(AbstractUser):

#class User(AbstractBaseUser, PermissionsMixin):

    #username_validator = MyValidator()

    GENDER_STATUS_CHOICES = (
```

```
    ('Male','Male'),
    ('Female','Female'),
    ('Others','Others')
)
PROGRAM_CHOICES = (
    ('Undergraduate', 'Undergraduate'),
    ('Post Graduate','Post Graduate'),
    #('Service Delivery','Service Delivery'),
)
PROFESSION_CHOICES = (
    ('Doctor', 'Doctor'),
    ('Nurse','Nurse'),
    ('Teacher','Teacher'),
    ('Civil Engineer','Civil Engineer'),
    ('Software Development','Software Development'),


    #('Service Delivery','Service Delivery'),
)
EXPERIENCE_CHOICES = (
    ('1-2 Years', '1-2 Years'),
    ('2-3 Years','2-3 Years'),
    #('Service Delivery','Service Delivery'),
)
#EXPERIENCE_CHOICES
    #('Service Delivery','Service Delivery'),
```

```
    is_client = models.BooleanField(default=False)

    phone_number = models.CharField(max_length=13)

    address = models.CharField(max_length=100, default="Kenya")

    #username = models.CharField( max_length=50,unique= True)

        is_account_active = models.BooleanField(default=False)

########################################################################################################################

    age = models.CharField(max_length=10)

    profession = models.CharField(max_length=20, blank=False, choices=PROFESSION_CHOICES)

    program = models.CharField(max_length=30,choices=PROGRAM_CHOICES)

    years_of_experience = models.CharField(max_length=30,choices=EXPERIENCE_CHOICES)

    gender =  models.CharField(max_length=30,choices=GENDER_STATUS_CHOICES)

    Country = models.CharField(max_length=20, blank=True)

    university = models.CharField(max_length=20,blank=True)


    description = models.TextField(max_length=800)

    year_of_graduation = models.DateField(default=timezone.now)

    profile_photo = models.ImageField(upload_to=cover_upload_path, blank=True,
default='clients/None/IMG_20190831_123316_G2XdrKx.jpg ')

    curiculum_vitae = models.FileField(upload_to=cover_upload_path, blank=True,
default='clients/None/IMG_20190831_123316_G2XdrKx.jpg ')

class Selections(models.Model):

    student = models.CharField(max_length=30)

    email = models.CharField(max_length=30)

    company = models.CharField(max_length=30)

    advert = models.CharField(max_length=30)

    date_selected = models.DateField(default=timezone.now)
```

```
    def __str__(self):
        return "%s" % (self.student)
class Application(models.Model):
    company = models.CharField(max_length=30)
    email = models.CharField(max_length=30)
    student = models.CharField(max_length=30)
    student_email = models.CharField(max_length=30)
    advert = models.CharField(max_length=30)
    description = models.TextField(max_length=1000)
    resume = models.FileField(upload_to=cover_upload_path, blank=True,
default='clients/None/IMG_20190831_123316_G2XdrKx.jpg ')
    date_selected = models.DateField(default=timezone.now)
class Feedback(models.Model):
    COMMENT_CHOICES = (
        ('Spam', 'Spam'),
        ('Company','Company'),
        ('Service Delivery','Service Delivery'),
        ('Others','Others')
    )
    user = models.ForeignKey(User,related_name='user_feedback',on_delete=models.CASCADE)
    topic = models.CharField(max_length=30,choices=COMMENT_CHOICES)
    body = models.CharField(max_length=300)
    publish_date = models.DateField(default=timezone.now)
    def __str__(self):
```

```
        return "%s" % (self.topic)
class NewsLetterUser(models.Model):
    email = models.EmailField()
    date_added = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.email
class NewsLetter(models.Model):
    EMAIL_STATUS_CHOICES = (
        ('Draft', 'Draft'),
        ('Published','Published')
    )
    subject = models.CharField(max_length=250)
    body = models.TextField()
    email = models.ManyToManyField(NewsLetterUser)
    status = models.CharField(max_length=10,choices=EMAIL_STATUS_CHOICES)
    created = models.DateTimeField(auto_now_add= True)
    updated = models.DateTimeField(auto_now =True)
class CompanyView(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    company = models.ForeignKey('Company', on_delete=models.CASCADE)
    def __str__(self):
        return self.user.username
class Comment(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    timestamp = models.DateTimeField(auto_now_add=True)
    content = models.TextField(null=True,blank=True)
```

company = models.ForeignKey('Company', related_name='comments', on_delete=models.CASCADE)

company_rating = models.PositiveIntegerField(default=0, validators=[MinValueValidator(1), MaxValueValidator(5)])

class Comment_reply(models.Model):

user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)

timestamp = models.DateTimeField(auto_now_add=True)

content = models.TextField(null=True,blank=True)

reply = models.ForeignKey('Comment', related_name='comment_reply', on_delete=models.CASCADE)

Appendix 2: Project time plan

No	**Task**	**Activities**	**Duration**
1	Review project scope and objectives	Review scope, objectives and measures of effectiveness in meeting them	**Week one**
		Establish project authority in relation to literature review	
2	Identify / review project infrastructure	Identify installation standards and procedures	**Week two**
		Analyze other project characteristics	
3	Analyze project characteristics	Identify high-level project risks	
		Consider general requirements concerning implementation	**Week three**
		Review and proposed general System life cycle approach	

No	Task	Activities	Duration
		Review proposed overall resource estimates	
		Identify and describe project end products (including quality criteria)	
		Review Documented generic product flows	
4	Review Identified project products and activities	Review proposed product instances	
		Review proposed activity network	
		Revise proposed plan to create controllable activities	

		Review Identified and quantified activity-based risks	**Week three**
		Review planned risk reduction and contingency measures where appropriate	
5	Review Identified	Revise plan and estimates to take account of	**Week three**

No	**Task**	**Activities**	**Duration**
	activity risks	resource constraints	
		Review quality aspects of project plan	
		Document plan	

		Document revised plan and align with general system development requirement	
	Analysis of Software Requirements		
6	Requirement Elicitation	Review project requirements, Justification and scope and expected outcomes	
7	System requirement Analysis and review	Establish system requirements	
		Establish and maintain traceability between project requirement and proposed conceptual framework	
		Verify system requirements for correctness, completeness, consistency and feasibility	**Week Four**

No	**Task**	**Activities**	**Duration**
		Draft preliminary software specifications	
		Baseline and document system requirements	
	Design		

8	Software Design	Review preliminary software specifications	Week Five
		Describe software architecture	
		Define interfaces	
		Develop detailed design	
		Develop prototype based on functional specifications	
		Establish and maintain traceability between software requirement and design	
		Review functional specifications	
		Incorporate feedback into functional specifications	
		Verify design for correctness, completeness, consistency and feasibility	

No	**Task**	**Activities**	**Duration**
		Baseline and align the design with requirements	
		Obtain approval to proceed	

	Development		
9	Software Construction (Coding)	Review functional specifications	**Week Six**
		Identify modular/tiered design parameters	
		Develop software units	
		Test software units (primary debugging)	
		Establish and maintain traceability between software requirement, design and software units	
		Verify code and test documents	
	Integration		
10	Software integration	Develop software integration strategy	
		Develop test for software item	

No	**Task**	**Activities**	**Duration**

		Integrate software item	
		Test integrated software	**Week Seven**
		Establish and maintain traceability between software design and software items	
		Verify integration	
		Test integrated software product	
11	System Integration	Develop system integration strategy	
		Develop tests for system elements	
		Integrate system elements	

		Test system elements	**Week Eight**
		Establish and maintain traceability between system design and software elements	
		Build completed system	
	Testing		

No	Task	Activities	Duration
12	Software Testing	Develop unit test plans using product specifications	
	(Unit Testing)		
		Review modular code	

		Test component modules to product specifications	
		Identify anomalies to product specifications	**Week Nine**
		Modify code	
		Re-test modified code	
		Unit testing complete	
13	System Testing	Develop integration test plans using product specification	
	(Integration Testing)		
		Test module integration	**Week Ten**

		Identify anomalies to product specifications	
		Modify code	

No	Task	Activities	Duration
		Re-test modified code	
		Integration testing complete	
	Training		
14	Training Materials	Develop training specifications for end users	
		Develop training specifications for support staff	
		Identify training delivery methodology	**Week Eleven**
		Develop training materials	
		Conduct training usability study	
		Finalize training materials	

		Develop training delivery mechanism	
15	Documentation	Develop Help specifications	
		Develop Help System	
		Review Help documentation	
		Incorporate Help documentation feedback	**Week twelve**

No	**Task**	**Activities**	**Duration**
		Develop user manual specifications	
		Develop user manuals	
		Review all user documentations	
		Documentation complete	
	Pilot		
16	Pilot System	Identify test group	
		Develop software delivery mechanism	
		Install/deploy software	**Week Thirteen**

		Obtain user feedback	
		Evaluate testing information	
		Pilot complete	
	Deployment		
17	System installation	determine installation strategy	
		Develop deployment methodology	
No	**Task**	**Activities**	**Duration**
		Deploy software product	
		Deployment complete	
18	Configuration management	Process implementation	
		Configuration identification	**Week Fourteen**
		Configuration control	
		Configuration status accounting	
		Configuration Audit	
		Release management and delivery	