Betr Selektr: A University Program Recommender System Utilising Personality Type and Academic Results

Belinda Ndlovu, National University of Science and Technology, Zimbabwe
Zirah Takunda Migioni, National University of Science and Technology, Zimbabwe
Sibusisiwe Dube, National University of Science and Technology, Zimbabwe
Phillip Nyoni, National University of Science and Technology, Zimbabwe

The Barcelona Conference on Education 2023
Official Conference Proceedings

Abstract
The goal of this system is to empower first-year students to make well-informed decisions about their university programs by providing tailored recommendations based on their individual profiles. Selecting the right university program can be a daunting task for first-year students. In response to this challenge, we have developed Betr Selektr, an all-encompassing program recommender system. This innovative system considers both the student's personality type and academic achievements, utilizing the Myers-Briggs Type Indicator (MBTI) framework. By merging this information with the student's high school results, we create a personalized index figure that reflects their unique personality type. This index figure acts as the foundation for recommending degree programs that align with the student's interests, strengths, and educational background. This system has been designed using the waterfall development methodology, employing tools such as Visual Studio Code, SQLAlchemy, Flask, and SQLite. Through various stages, including systems analysis and design, implementation and testing, and the utilization of research methodologies, we have created a comprehensive solution. Betr Selektr offers a user-friendly interface, swift data processing, and precise program recommendations, making it an invaluable asset in the university application process.

Keywords: Betr Selektr, University Program, Myers-Briggs Type Indicator (MBTI) Framework

I. Introduction

The process of selecting a suitable university program is a critical decision for students entering their first year of higher education. To assist students in making informed choices, university program recommender systems have emerged as valuable tools (Denley, 2012). These systems leverage advanced technologies and data analytics techniques to provide personalized recommendations based on various factors, including academic performance, interests, and personality traits. One such innovative recommender system is Betr Selektr, which combines the use of personality type assessment and academic results to recommend degree programs tailored to individual students. This research aims to develop a system namely Betr Selektr, to support first-year students' decision-making process when selecting a degree program.

Literature reviews have highlighted the significance of personalized recommendations in the university program selection process. The ability to match a student's unique characteristics, strengths, and interests with appropriate degree programs enhances their satisfaction, engagement, and overall academic success (Kemboi et al, 2016). Furthermore, research studies have shown the influence of personality traits on academic performance and career outcomes (Tucker et al, 2016). Incorporating personality type assessment into the recommendation process can provide valuable insights into students' preferences, learning styles, and future aspirations. Betr Selektr adopts the well-established Myers-Briggs Type Indicator (MBTI) framework as a means to assess students' personality types. The MBTI classifies individuals into specific personality dimensions, including extraversion/introversion, sensing/intuition, thinking/feeling, and judging/perceiving. By combining the personality type assessment with academic results, Betr Selektr generates a personalized index figure that serves as the basis for recommending degree programs aligned with the student's profile.

A. Aim and Objectives

To design a system that recommends which university programs a student applicant should apply for considering their personality type as the primary criteria.

The objectives are as follows:

- To create a dataset of the 16 personality types
- To create a dataset of the programs offered by the National University of Science and Technology.
- To create a knowledge base containing the personality types, programs as well as the Sixth Form Second and Final Term academic results of prospective applicants.
- To predict which programs an applicant should choose in line with their personality type but controlled by their Sixth Form academic results.
- To recommend which programs a NUST applicant should put on their application form.

II. Literature Review

We identified two (2) main techniques employed to run recommendation systems.

1. Collaborative Filtering (CF). This is a method of filtering that focuses on the relationships between users and items. (Majidi, 2018)

2. Content-Based Filtering: This is a method of filtering that focuses on the property of items (Shahab, 2019)

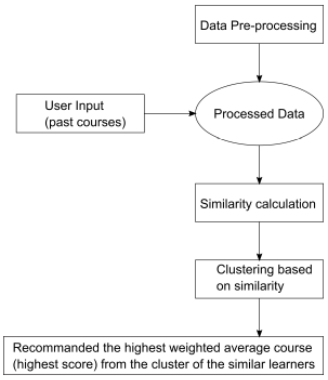

Fig. 1. Collaborative Filtering

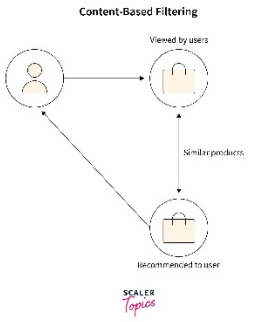

Fig. 2. Content Based Filtering

Collaborative filtering is a technique that relies on user behaviour data, such as past preferences or ratings, to make recommendations (Ricci et al, 2010). It identifies patterns and similarities among users based on their interactions with items or services. The basic idea behind collaborative filtering is that users who have similar preferences in the past are likely to have similar preferences in the future. This approach does not rely on explicit item attributes but rather on the behaviour and preferences of users themselves. It can be further categorized into two types: user-based collaborative filtering and item-based collaborative filtering (Su et al, 2009).

On the other hand, content-based filtering focuses on the characteristics or attributes of items themselves to make recommendations (Mondal et al, 2020). It analyses the features or content of items and matches them with the user's preferences or profile. For example, in the context of movie recommendations, content-based filtering would consider attributes such as genre, director, actors, and plot summaries to find similar movies based on the user's preferences for these attributes. This approach does not rely on user behaviour data but rather on the item's intrinsic features.

A. Limitations of Existing Systems

1. ***Based on Grades***: Mondal et al (2020) developed a course recommendation system based on student grades. The very nature of student grades entails the need to use historical and survey data, which is what the researchers endeavoured to use. Mondal's system would classify the learners using the historical data previously

alluded to by finding out the background of the students who managed to attain a higher mark or grade in each course attempted. Each time the system logged a new learner they would be classified according to the system's existing clusters and as a result, a set of recommended courses would then be availed to the user based on the frequent pattern mining algorithm.

Mondal (2020) noted that the personalization of course recommender systems is lagging. Those that exist use content filtering or collaborative filtering and frequent pattern mining without taking a personal approach to users. This is a gap Betr Selektr aims to fill by proposing a personalized experience. In as much as Betr Selektr is proposing to use a variant of Collaborative Filtering, it endeavors to hybridize this process.

2. ***Based on Fuzzy Logic***: The study carried out by Sulaiman et al (2020), notes that, "a key to a student's success in tertiary education is choosing the right course and the need for a deep interest in each course they would have chosen." Upon realising the inadequacies of the different systems, they had studied, they decided to use a fuzzy logic approach whilst including multivariate questioning techniques in order to combat or rather fill the gap left by previous researchers. Fuzzy Logic endeavours to mimic human intelligence in solving a specific problem, this is a subset of the broader field of Artificial Intelligence (K. Tanaka 1996 cited in Mondal et al 2019).

   Despite the relevance of the study carried out by Sulaiman et al (2019), it was heavily constrained by the narrowed focus and specialisation on just computer science-related courses. Thus, Betr Selektr is looking to recommend courses or programmes across the board without a heavy reliance and or focus on one field.

3. ***Based on Career Goals***: Narges Majidi (2018) designed a course recommender system using career goals as the main basis of its recommendations. The system uses a variety of data mining algorithms simultaneously to enhance the accuracy of the recommendations controlled by the student's career goals. Some of the algorithms they incorporated into their system are the Apriori Algorithm, Greedy algorithm and the genetic algorithm.

   The system's shortfall is in its limiting factor, the career choice path. It markedly neglects the pre-existing inbred reasoning why people have those career goals that they have and what pushes them to attain those goals. In the event that the push or motivating factor changes, the career goal will change as well, however personality tends to be a more reliable basis. This makes it such that its reliance on such an unstable and often superficial factor prone to changes.

4. ***Based on Graduating Attributes***: Behdad Bankshinategh, et al., (2017) designed a course recommender system based on graduating attributes. It assesses a student's competencies and assigns a course based on those competencies. Within academia, competencies are "multidimensional constructs composed of the skills, attitudes, and behaviours of a learner that contribute to academic success in the classroom" (DiPerna and Elliott, 1999).

   The graduating attributes were not limited to just the academic aspect of learning but the deep intrapersonal component of what it takes to actually pass courses. There is a

need for intrapersonal factors in the realm of course recommendation so as to recommend that which speaks to individuals at a personal level.

III. Methodology

A. Research Methodology

The software development methodology chosen for this course recommendation project is the waterfall model. The waterfall model is a sequential development approach that emphasizes a linear and structured approach to software development. It involves several phases, including requirements gathering, design, implementation, testing, and maintenance. Each phase must be completed before proceeding to the next phase. This methodology has been widely used in various software development projects due to its well-structured and systematic approach (Adobe Systems, 2022).

After considering the various options, the waterfall development model was selected as the most appropriate approach for this project. The primary reason for this decision is that the requirements for the system are well-defined and the steps required to build the system can be clearly identified and followed in a sequential manner. The waterfall model is particularly well-suited for projects that have a clear set of requirements and a defined set of steps, as it allows for a structured and organized approach to development. This is important in the context of this project because it allows the team to focus on one stage of development at a time, which helps to ensure that the project stays on track and stays within budget. Additionally, the waterfall model allows for thorough testing at each stage of the development process, which is important for ensuring the quality and reliability of the final product. This is especially important in the context of a course recommendation system, as it is essential that the system provides accurate and relevant recommendations to students (Sherman , 2015).

While the agile and lean development methodologies may be suitable for other types of projects, they are less well-suited for this project because they prioritize flexibility and adaptability over structure and organization. While these qualities can be beneficial in some contexts, they are not as important in the context of this project, which has well-defined requirements and a clear set of steps to be followed.

Overall, the waterfall development model is the most appropriate.

IV. Design and Implementation

A. System Modelling

We used the Unified Modelling Language tools to map out what the system would look like. The Use Case diagram below indicates the main activities of the system from the perspective of the students:

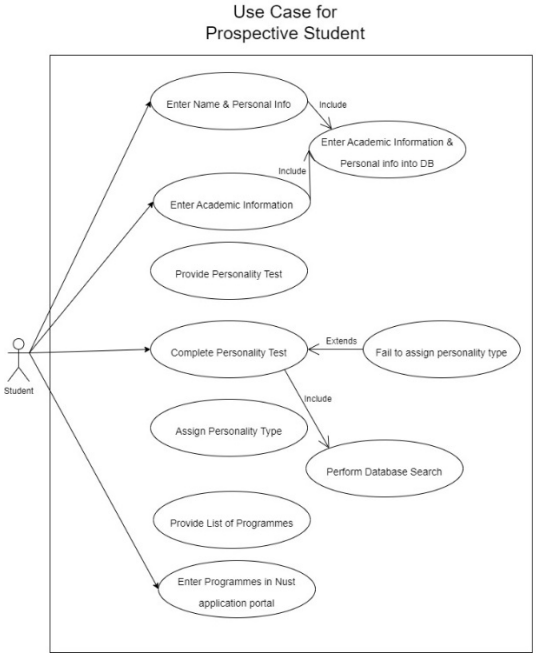

Fig. 3. Use Case Diagram

The figure below shows the flow of activities in a Sequence Diagram:

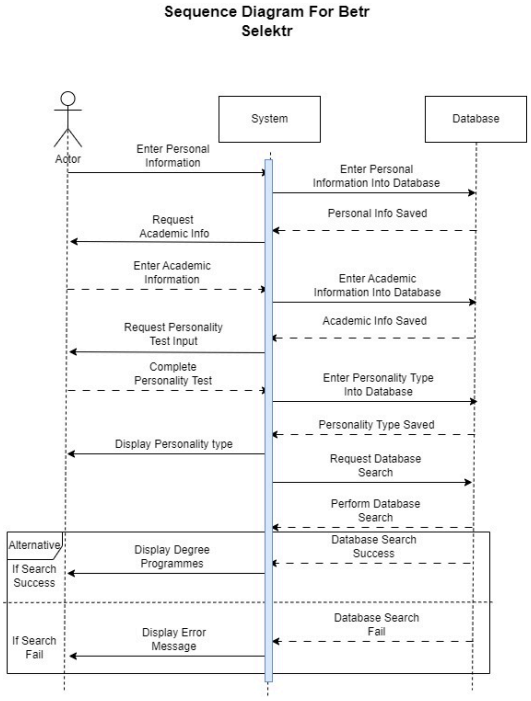

Fig. 4. Sequence Diagram for Betr Selektr

B. User Interface Design

The user interface was designed using HTML via Python's Flask Library.

V. Implementation

A. Code Snippets

The deployment of this system was largely dependent on leveraging web application technology. We used the Python's lightweight web app development library – Flask. Below we will demonstrate key code snippets that show how the web app was implemented and deployed.

Main.py

The main.py module contains the code with the root of the app. It essentially is where the app will be stored:

```
from website import create_app

app = create_app()

if __name__ == '__main__':
        app.run(debug=True)
```

Init.py

The next key module is the init.py module which contains the initialisation of Betr Selektr's modules, objects, classes and routes. This module also creates and initialises the SQLAlchemy database as well as managing its connections with the broader system, for example logins. The SQLAlchemy database is configured here and is run from this module specifically.

```
from flask import Flask

from flask_sqlalchemy import SQLAlchemy

from os import path

from flask_login import LoginManager


db = SQLAlchemy()

DB_NAME = "mydatabase.db"


def create_app():
    app = Flask(__name__)

    app.config['SECRET_KEY'] = 'Zirah08'

    app.config['SQLALCHEMY_DATABASE_URI'] = f'sqlite:///{DB_NAME}'

    app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
```

```
    db.init_app(app)
    from .views import views
    from .auth import auth

    app.register_blueprint(views, url_prefix='/')
    app.register_blueprint(auth, url_prefix='/')

    from .models import Programme, PersonalityType, User

    login_manager = LoginManager()
    login_manager.login_view = 'auth.login'
    login_manager.init_app(app)
    @login_manager.user_loader
    def load_user(id):
        return User.query.get(int(id))
    with app.app_context():
        db.create_all()
    return app
def create_database(app):
    if not path.exists('website/' + DB_NAME):
        db.create_all(app=app)
        print('Created Database!')
```

Auth.py (Personality Test)

This algorithm is the key to this whole project. To figure out which personality type an applicant has. This algorithm is what we have used to test and assign personality types to applicants. It is based on a simple structure of using a scale to assign which attribute a person exhibits more of for example; Extraversion and Introversion.

```
@auth.route('/personality_test', methods=['GET', 'POST'])
def personality_test():
    if request.method == 'POST':
        # Initialize scores for each attribute
```

```
scores = {'E': 0, 'I': 0, 'S': 0, 'N': 0, 'T': 0, 'F': 0, 'J': 0, 'P': 0}
#calculate the personality score
for i in range(1, 9):
    answer = int(request.form['circleE{}'.format(i)])
    scores['E'] += answer
    scores['I'] += 6 - answer

for i in range(1, 9):
    answer = int(request.form['circleS{}'.format(i)])
    scores['S'] += answer
    scores['N'] += 6 - answer

for i in range(1, 9):
    answer = int(request.form['circleT{}'.format(i)])
    scores['F'] += answer
    scores['T'] += 6 - answer

for i in range(1, 9):
    answer = int(request.form['circleJ{}'.format(i)])
    scores['J'] += answer
    scores['P'] += 6 - answer

personality_type = ''

if scores['E'] > 20:
    personality_type += 'E'
else:
    personality_type += 'I'
```

```
if scores['S'] > 20:
    personality_type += 'S'
else:
    personality_type += 'N'

if scores['T'] > 20:
    personality_type += 'T'
else:
    personality_type += 'F'

if scores['J'] > 20:
    personality_type += 'J'
else:
    personality_type += 'P'

flash('Your personality type is: {}'.format(personality_type))

# Retrieve personality description from the database
personality = PersonalityType.query.filter_by(type=personality_type).first()
if personality:
    personality_description = personality.description
else:
    personality_description = ''

# Retrieve programs from the database
programs = Programme.query.filter_by(personality_type=personality_type).all()
```

```
    if current_user.is_authenticated:
        user = User.query.filter_by(id=current_user.id).first()
        user.user_personality = personality_type
        db.session.commit()

    # Retrieve personality description from the database
    personality = PersonalityType.query.filter_by(type=personality_type).first()
    if personality:
        personality_description = personality.description
    else:
        personality_description = ''

    # Retrieve programs from the database
    programs = Programme.query.filter_by(personality_type=personality_type).all()


    return render_template('personality_test.html', user=current_user, personality_description=personality_description, programs=programs)


@auth.route('/test_results')
@login_required
def test_results():
    user_personality = current_user.user_personality
    personality_type = PersonalityType.query.filter_by(personality_type=user_personality).first()
    description = personality_type.description
    programs = personality_type.programs
    return render_template("home.html",user=current_user, personality_type=user_personality, description=description, programs=programs)
```

The personality test will generate the personality type used to query the database.

B. Screenshots

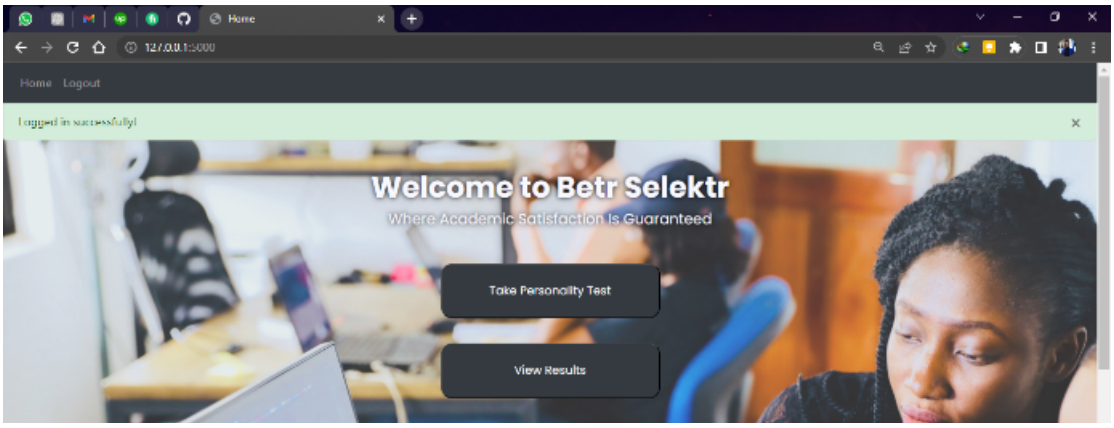

Fig. 5. Home Page

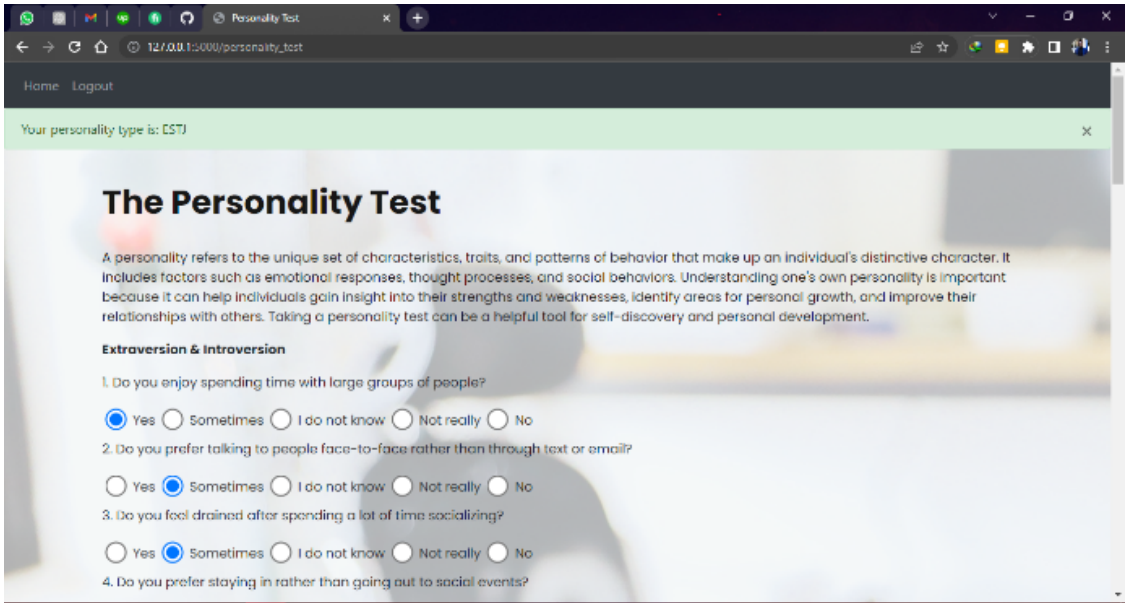

Fig. 6. Personality Test

The personality test is an HTML form with 32 questions divided into the 4 categories that measure a person's personality type as devised by Myers and Briggs. The categories are: 1. Extraversion vs Introversion 2. Sensing vs Intuition 3. Thinking vs Feeling 4. Judging vs Perceiving Each category has 8 questions and each question has 5 radio buttons with assigned values which are used for computing which personality attribute a user has. This is computed by the auth.py module under the route module that handles the submission of the forms data.

In the example above, the user scored an ESTJ personality type and is then redirected to the home page so that they can see their recommendations as well as additional information pertaining to their personality type.

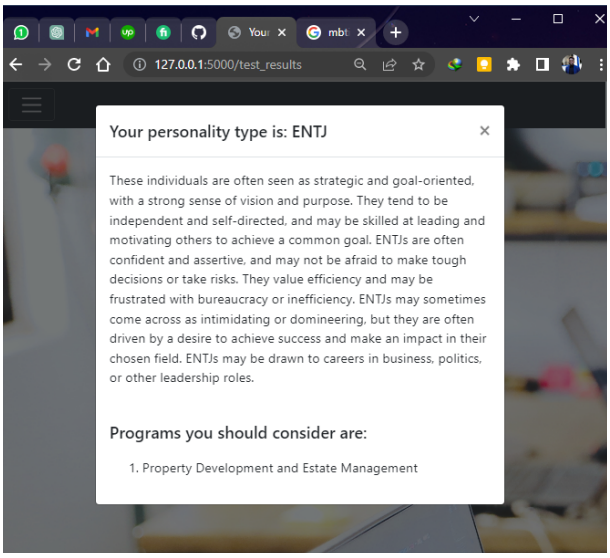

Fig. 7. Results

C. Limitations of the System

The system relies heavily on a personality test that was built by our team with the help of an expert.

The system also in some isolated cases only recommends one programme which does not really give the applicant much choice.

VI. Conclusion

After analysis, design, and implementation, Betr Selektr, a university program recommender system that utilizes personality type and academic results to recommend suitable courses, has been successfully developed. The system was developed using the Waterfall software development methodology, which enabled a structured and sequential approach to the project. The system was built using Python programming language and various technologies such as Visual Studio Code, SQLAlchemy, and Flask. The implementation phase involved building the system's database, creating the user interface, integrating the personality test, and testing the system thoroughly. The system demonstrated its ability to accurately recommend courses based on user input.

In conclusion, the project was successful in achieving its objectives and creating a useful tool for first-year university students to navigate the overwhelming process of choosing courses. Future work on the project could involve expanding the personality test questions to increase accuracy, incorporating additional criteria such as extracurricular activities, and improving the user interface. The project serves as a valuable demonstration of the application of software development methodologies and technologies in solving real-world problems.

References

Adobe Communication Team (2022) *Waterfall Methodology: A Complete Guide.* https://business.adobe.com/blog/basics/waterfall

Bakhshinategh, Behdad & Spanakis, Gerasimos & Zaïane, Osmar & Elatia, Samira. (2017). A Course Recommender System based on Graduating Attributes. 347-354. 10.5220/0006318803470354

Denley, T. (2012). Austin Peay State University: Degree Compass. In D. G. Oblinger (Ed), Game Changers: Education and Information Technologies (pp. 263-267). Washington, DC: Educause. http://net.educause.edu/ir/library/pdf/pub7203.pdf

Hollweck, Trista. (2016). Robert K. Yin. (2014). Case Study Research Design and Methods (5th ed.). Thousand Oaks, CA: Sage. 282 pages. The Canadian Journal of Program Evaluation. 30. 10.3138/cjpe.30.1.108.

Kemboi R.J.K, Kindiki Nyaga, Misigo Benard (2016): Relationship between Personality Types and Career Choices of Undergraduate Students: A Case of Moi University, Kenya

Mohd Suffian Sulaiman, Amylia Ahamad Tamizi, Mohd Razif Shamsudin, Azri Azmi (2020): Course recommendation system using fuzzy logic approach.

Mondal, Bhaskar & Patra, Om & Mishra, Sanket & Patra, Priyadarsan. (2020). A course recommendation system based on grades. 1-5. 10.1109/ICCSEA49143.2020.9132845.

Narges Majidi (2018): A Personalized Course Recommendation System Based on Career Goals

Ponto, Julie. (2015). Understanding and Evaluating Survey Research. Journal of the advanced practitioner in oncology. 6. 168-171.

Reason, P., & Bradbury, H. (Eds.). (2013). The SAGE handbook of action research: Participative inquiry and practice. Sage

Ricci, Francesco & Rokach, Lior & Shapira, Bracha. (2010). Recommender Systems Handbook. 10.1007/978-0-387-85820-3_1.

Rick Sherman,(2015) Chapter 18 - Project Management, Pages 449-492, ISBN 9780124114616, https://doi.org/10.1016/B978-0-12-411461-6.00018-6

Shadish, William & Hedges, Larry & Pustejovsky, James. (2014). Analysis and meta-analysis of single-case designs with a standardized mean difference statistic: A primer and applications. Journal of school psychology. 52. 123-147. 10.1016/j.jsp.2013.11.005.

Shehba Shahab (2019): NEXT LEVEL: A COURSE RECOMMENDER SYSTEM BASED ON CAREER INTERESTS

Su, Xiaoyuan & Khoshgoftaar, Taghi. (2009). A Survey of Collaborative Filtering Techniques. Adv. Artificial Intelligence. 2009. 10.1155/2009/421425.

Tucker RP, Lengel GJ, Smith CE, Capron DW, Mullins-Sweatt SN, Wingate LR (2016). Maladaptive Five Factor Model personality traits associated with Borderline Personality Disorder indirectly affect susceptibility to suicide ideation through increased anxiety sensitivity cognitive concerns. Psychiatry Res 246:432-437. doi:10.1016/j.psychres.2016.08.051. Epub. PMID: 27788465.

Contact emails: belinda.ndlovu@nust.ac.zw
zirahmigioni@gmail.com
sibusisiwe.dube@nust.ac.zw
phillip.nyoni@nust.ac.zw