COURSE RECOMMENDATION SYSTEM

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

PENUGONDA RAJASEKHAR REDDY	(21UECS0466)	**(VTU19381)**
PITTU DEVENDAR REDDY	(21UECS0477)	**(VTU19934)**
SATTARU HARSHA VARDHAN REDDY	(21UECS0553)	**(VTU20580)**

Under the guidance of
Mr. SANKAR GANESH. K, M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

COURSE RECOMMENDATION SYSTEM

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

PENUGONDA RAJASEKHAR REDDY	(21UECS0466)	**(VTU19381)**
PITTU DEVENDAR REDDY	(21UECS0477)	**(VTU19934)**
SATTARU HARSHA VARDHAN REDDY	(21UECS0553)	**(VTU20580)**

Under the guidance of
Mr. SANKAR GANESH. K, M.E.,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”COURSE RECOMMENDATION SYSTEM” by ”PENUGONDA RAJASEKHAR REDDY (21UECS0466), PITTU DEVENDAR REDDY (21UECS0477), SATTARU HARSHA VARDHAN REDDY (21UECS0553)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor	**Signature of Professor In-charge**
Computer Science & Engineering	**Computer Science & Engineering**
School of Computing	**School of Computing**
Vel Tech Rangarajan Dr. Sagunthala R&D	**Vel Tech Rangarajan Dr. Sagunthala R&D**
Institute of Science & Technology	**Institute of Science & Technology**
May, 2024	**May, 2024**

DECLARATION

We declare that this written submission represents my ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

Penugonda Rajasekhar Reddy

Date: / /

Signature

Pittu Devendar Reddy

Date: / /

Signature

Sattaru Harsha Vardhan Reddy

Date: / /

APPROVAL SHEET

This project report entitled "COURSE RECOMMENDATION SYSTEM" by PENUGONDA RAJASEKHAR REDDY (21UECS0466), PITTU DEVENDAR REDDY (21UECS0477), SATTARU HARSHA VARDHAN REDDY (21UECS0553) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	Mr. Sankar Ganesh. K, M.E.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E, Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mr. SANKAR GANESH. K, M.E.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr.V. ASHOK KUMAR, M.Tech., Ms.U. HEMAVATHI, M.E., Ms.C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

PENUGONDA RAJASEKHAR REDDY (21UECS0466)
PITTU DEVENDAR REDDY (21UECS0477)
SATTARU HARSHA VARDHAN REDDY (21UECS0553)

ABSTRACT

In today's world, choosing the right online course from the myriad options available can be challenging. This project introduces a sophisticated Course Recommendation System (CRS) designed to assist individuals in discovering courses aligned with their interests. Imagine a system that functions like a knowledgeable friend suggesting books based on the user preferences, but in this case, it recommends online courses. The CRS employs two distinct approaches: one that quickly identifies similar courses, suitable for a smaller collection, and another that comprehensively understands course content for more accurate recommendations, especially in larger datasets.The system uses k-NN machine learning algorithm and the TF-IDF vectorizer to vectorize the text.The system employs a combination of linguistic and numerical analyses to grasp course attributes, aiming to enhance the accuracy of recommendations. Using specialized algorithms, the system seeks to improve user satisfaction with selected courses and promote academic success.To make the CRS work, the system initially gathers information from users about their favourite subjects and preferred learning methods. This data is then used to build an intelligent system that can accurately recommend courses. By employing mathematical and computational techniques, the system transforms user preferences into a functional CRS prototype. This tool proficiently suggests courses tailored to individual preferences. Continuous improvements are made based on user feedback and rigorous testing, ensuring ongoing usefulness for the broader community.

Keywords:
Course Recommender, Online Learning, k-Nearest Neighbors (k-NN), TF-IDF, Recommendations, Text Analysis, Machine Learning, Vectorization, Data Processing, User Preferences.

LIST OF FIGURES

4.1 Architecture Diagram for Course Recommendation System . . . 11
4.2 Data Flow Diagram for Course Recommendation System 12
4.3 Use Case Diagram for Course Recommendation System 13
4.4 Class Diagram for Course Recommendation System 14
4.5 Sequence Diagram for Course Recommendation System 15
4.6 Collaboration Diagram for Course Recommendation System . . 16
4.7 Activity Diagram for Course Recommendation System 17
4.8 Architecture Diagram for Data Collection and Data Processing . 19
4.9 Architecture Diagram for Vectorization Module 20
4.10 Architecture Diagram for KNN Algorithm 21
4.11 Raw data 22
4.12 Processed data 23

5.1 Input design of Course Recommender 24
5.2 Output design of Course Recommender 24
5.3 Unit Test Input 25
5.4 Unit Test Output 25
5.5 Integration Test Input 26
5.6 Integration Test Output 26
5.7 System Test Output 27
5.8 System Testing Result 28

6.1 Output 32

8.1 Plagiarism 35

9.1 Poster 38

LIST OF ACRONYMS AND ABBREVIATIONS

API	Application Programming Interface
AWS	Amazon Web Services
CRS	Course Recommendation System
CORS	Cross Origin Resource Sharing
CSV	Comma Separated Values
GB	Giga Byte
GPU	Graphical Processing Unit
IDE	Integrated Development Environment
IP	Internet Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbors
ML	Machine Learning
PEP	Python Enhancement Proposa
RAM	Random Access Memory
SSD	Solid State Disk
TF-IDF	Term Frequency- Inverse Document Frequency
UI	User Interface

TABLE OF CONTENTS

Page.No

ABSTRACT **v**

LIST OF FIGURES **vi**

LIST OF ACRONYMS AND ABBREVIATIONS **vii**

1 INTRODUCTION **1**
- 1.1 Introduction 1
- 1.2 Aim of the project 1
- 1.3 Scope of the Project 2

2 LITERATURE REVIEW **3**

3 PROJECT DESCRIPTION **7**
- 3.1 Existing System 7
- 3.2 Proposed System 7
- 3.3 Feasibility Study 8
  - 3.3.1 Economic Feasibility 8
  - 3.3.2 Technical Feasibility 8
  - 3.3.3 Social Feasibility 9
- 3.4 System Specification 10
  - 3.4.1 Hardware Specification 10
  - 3.4.2 Software Specification 10
  - 3.4.3 Standards and Policies 10

4 METHODOLOGY **11**
- 4.1 Course Recommender Architecture 11
- 4.2 Design Phase 12
  - 4.2.1 Data Flow Diagram 12
  - 4.2.2 Use Case Diagram 13
  - 4.2.3 Class Diagram 14
  - 4.2.4 Sequence Diagram 15

4.2.5 Collaboration diagram 16
4.2.6 Activity Diagram 17
4.3 Algorithm & Pseudo Code 18
4.3.1 Vectorization with TF-IDF and Classification using K-Nearest Neighbors 18
4.3.2 Pseudo Code 18
4.4 Module Description 19
4.4.1 Module 1: Data collection and Processing 19
4.4.2 Module 2: Vectorization (TF IDF vectorizer) 20
4.4.3 Module 3: KNN Algorithm 21
4.5 Steps to implement the project 22
4.5.1 Step 1: Data Collection and Processing 22
4.5.2 Step 2: Building the model and the back end using flask . . 23
4.5.3 Step 3: UI designing using Flutter 23

5 IMPLEMENTATION AND TESTING 24
5.1 Input and Output 24
5.1.1 Input Design: Course Recommendation System 24
5.1.2 Output Design: Course Recommendation System 24
5.2 Testing 25
5.3 Types of Testing 25
5.3.1 Unit testing 25
5.3.2 Integration testing 26
5.3.3 System testing 27
5.3.4 Test Result 28

6 RESULTS AND DISCUSSIONS 29
6.1 Efficiency of the Proposed System 29
6.2 Comparison of Existing and Proposed System 29
6.3 Sample Code 30

7 CONCLUSION AND FUTURE ENHANCEMENTS 33
7.1 Conclusion 33
7.2 Future Enhancements 33

8 PLAGIARISM REPORT 35

9 SOURCE CODE & POSTER PRESENTATION **36**
9.1 Source Code 36
9.2 Poster Presentation 38

References **38**

Chapter 1

INTRODUCTION

1.1 Introduction

”Ever felt overwhelmed by the countless educational courses available? Imagine a world where choosing the perfect course is as easy as a personalized recommendation”.This project revolves around the development of a Course Recommendation System (CRS) to revolutionize the course selection process. Harnessing the power of advanced algorithms and user input, the CRS aims to tailor course suggestions based on individual preferences .The project recognizes the challenge of navigating a vast educational landscape and seeks to simplify this journey for users. With a user-centric approach, the CRS endeavors to enhance satisfaction and success by providing a dynamic, personalized, and efficient course recommendation platform.

The system continuously refines its recommendations based on user feedback and interactions, ensuring adaptability to evolving learning preferences. This innovative application of machine learning not only streamlines the course discovery process but also enhances the overall learning experience by aligning educational content with individual needs and interests.

The Course Recommendation System Project is an innovative application designed to assist users in discovering online courses aligned with their interests. By harnessing the power of machine learning, this system analyzes course descriptions to suggest relevant and engaging learning opportunities. Using a combination of TFIDF (Term Frequency-Inverse Document Frequency) vectorization and the k-Nearest Neighbors (k-NN) algorithm, it transforms textual information into numerical features and identifies courses with similarities.

1.2 Aim of the project

The aim of the course recommendation system project utilizing machine learning is to enhance the educational experience by providing personalized and tailored recommendations to learners. Through the analysis of user preferences, past academic performance, and learning styles, the system aims to suggest courses that align with individual needs and interests. The primary objective is to optimize the learning journey, ensuring that students receive relevant and engaging content, thereby increasing their motivation and overall academic success.

By leveraging machine learning algorithms, the system can continuously adapt and evolve its recommendations based on real-time user feedback and changing educational requirements. Ultimately, the goal is to create a more efficient and effective educational ecosystem that caters to the diverse learning preferences of students, facilitating their academic growth and fostering a lifelong love for learning.

1.3 Scope of the Project

The scope of a crop disease detection project is extensive, encompassing various dimensions of agricultural technology and data science. At its core, the project aims to address the critical challenge of timely disease identification in crops, thereby enhancing agricultural productivity and ensuring food security. The scope includes the development of robust algorithms and machine learning models capable of processing large datasets of crop images. These models can learn to differentiate between healthy and diseased crops based on visual cues, enabling early detection and intervention. The project may explore the integration of emerging technologies such as drones and satellite imagery to facilitate scalable and efficient monitoring of vast agricultural landscapes.

Furthermore, the scope extends to the implementation of a user-friendly interface that allows farmers to interact with the system seamlessly. This interface could take the form of a mobile application or a web platform, providing real-time insights into the health of their crops and offering actionable recommendations for disease management. Additionally, the project may involve the incorporation of educational components to empower farmers with knowledge about prevalent diseases, their causes, and potential preventive measures. By encompassing these facets, the project's scope aims not only to detect diseases but also to promote sustainable and informed agricultural practices, contributing to the overall well-being of the agricultural ecosystem.

Chapter 2

LITERATURE REVIEW

[1] B. Mondal et al., proposed a machine learning approach to recommend suitable courses to learners based on their learning history and past performance. The model first classifies a new learner based on their past performance using the k-means clustering algorithm. Collaborative filtering will be applied in the cluster to recommend a few suitable courses. Further, based on an online test the adaptability of the learner will be tested to the customized recommended courses according to learners needs. The framework will provide a personalized environment of study to each learner.

[2] Derick Prince B et al., discussed on the education system that has undergone a significant transformation due to technological advancements. With the emergence of online learning platforms, students can now access an enormous amount of course materials, resources, and lectures online. However, the abundance of options can also create confusion and make it difficult for students to decide on courses to be taken, especially when considering their career aspirations. To address this problem, the proposed system introduces an integrated course and job recommendation system that utilizes machine learning techniques like Na¨ıve Bayes and Collaborative filtering algorithms to suggest relevant courses and job opportunities based on the student's interests, skills, and career goals.

[3] H. Aoulad Ali et al., demonstrated a system that make recommendations One of the most intriguing systems, which allows students to browse or search a big number of possibilities for specific goods related to their interests. Using varied methodologies such as the content-based approach and the collaborative approach, these systems have assisted learners in e-learning in filtering options and selecting information. The main purpose of this proposed technique is to generate homogeneous groups of learners in order to cover and absorb all of the essential information.

[4] M. I. Emon et al., demonstrated a structured framework of an intellectual recommendation system for self-learners so that they can get right direction and a personalized learning road-map through recommendations. In recent years, Recommendation system has developed rapidly and used widely. In online educational system recommendation system can be used to provide educational resources to students that matches their interests of learning. Personalized recommendation will be provided to the students through mining data. Student's activity will be recorded and stored in profiles. They have an intelligent support system to support our learners to select their courses as per their educational qualifications.

[5] M. Isma'il et al., discussed on tackling the wrong placement of applicants into courses and to also address the wastage of admission vacancies by recommending the appropriate course(s). About 8,700 data for three academic sessions were collected from two different universities for training and testing the system. The features used are the results of Senior Secondary School subjects and the average score of UTME and Post-UTME. The proposed system employed five classification models, which include Linear Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbor and Decision Tree Algorithm.The result shows that both Naive Bayes Classifier and Support Vector Machine have achieved the highest recommendation accuracy of approximately 99.94 percentage, which outperformed Decision Tree and K-Nearest Neighbor algorithms with an accuracy of 98.10 percentage and 99.87 percentage respectively.

[6] M. M. Rahman et al., explained the difficulty students have choosing the most suitable courses during their undergraduate studies. In the academic world, the institutions offer students various courses to study. They have several options to choose from many courses based on their future career planning, interest, and advice from peers, seniors, teachers, etc. Hence, inappropriate course selection leads to innumerable difficulties, poor performance and dissatisfaction. Thus, it's essential to propose a course recommendation system that helps undergraduate students in the course selection process. This paper presents a machine learning approach to recommend relevant courses to students based on popular courses. The K-Means clustering method has been used to find students' most and least demand courses. Then, the FP-Growth algorithm generates the rules to recommend suitable courses for a specific student. A real-world dataset has been used, which consists of

undergraduate students' academic records. This proposed method is evaluated by applying the dataset that would perform relatively better.

[7] P. Mishra and V. Jain, explained about the E-learning infrastructure that is growing rapidly, choosing the right skills set to built a career in an area of interest sometimes can be mystifying and hence a recommendation system is helpful to narrow down the information or choices based on user's data or preferences. A recommender system automates the process of filtering and make it feasible for a user to search through vast information available online and thus provide a personalized experience for the user. This study has attempted to implement a recommender system based on content based filtering and Machine Learning algorithm to filter skills and courses available digitally based on user's input information.

[8] S. Lazarevic et al., proposed a machine learning-driven course recommendation system based on similarities between courses. The proposed system employs various data mining techniques to mentioned similarities between courses. Based on the experimental phase of this paper, Cosine metrics proved the best to calculate these parameters. The method proposed in this paper relies on rankings based on areas of study. These techniques allowed us to create an algorithm that, based on input, returns courses that satisfy various conditions. The results satisfy the demands of finding similar courses presented through cross-platform application to the students who will use it to improve their education.

[9] V. Sankhe et al., discussed on the overview of today's world where students face an immense repertoire of options relating to the number of courses that they may choose from. To make this seemingly massive choice relatively easy to make, many authors have created their own recommender systems to map students to the courses that are best suited for them. However, they are not widely used as they give good results only for the dataset that they consider. In this paper, they have mapped the current students to their alumni based on multiple criterion. Afterwards, unlike other papers that used k-means, they used c-means and fuzzy clustering to arrive at a better solution to predict an elective course for the student. Since all of this is done on a broad actual dataset, the results can be applied anywhere in a real world scenario.

[10] X. Pan et al., described a course recommendation model based on deep learning, which can generate views from a different perspective and make intelligent course recommendations for students. First, collect various types of data to generate student models and curriculum models. By studying the actual relationship between specific models, and then using deep learning technology to extract key features, select different recommended features from these relationships, and generate different views to further recommend for students

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The K-Nearest Neighbors (K-NN) can be effective for certain recommendation tasks, relying solely on this algorithm for a course recommender system may present several limitations:
1. Scalability: As the dataset grows, K-NN's computational requirements increase significantly because it needs to calculate distances to every data point. This can make the algorithm slow and resource-intensive, affecting its scalability.
2. Cold Start Problem: K-NN struggles with new or less popular courses since it relies on similarities between existing data points. New courses or those lacking sufficient data might not have enough neighbors for accurate recommendations (cold start problem).
3. Impact of Irrelevant Features: K-NN considers all features equally important. If irrelevant or noisy features are present in the dataset, they can negatively impact the algorithm's accuracy, leading to suboptimal recommendations.
4. No Training Phase: Unlike many machine learning algorithms, K-NN doesn't require an explicit training phase. It directly uses the entire dataset for predictions, making it simple and flexible.

3.2 Proposed System

The advantages of using vectorization (such as TF-IDF) and the K-Nearest Neighbors (K-NN) algorithm for a course recommendation system:
1. Text Representation: Vectorization methods like TF-IDF convert textual course descriptions into numerical representations, allowing algorithms to process and analyze text-based data effectively.
2. Feature Extraction: TF-IDF identifies important words or phrases in course descriptions, capturing essential information and reducing text complexity into mean-

ingful features.

3. Sparse Representation: TF-IDF creates a sparse matrix representation, which efficiently handles large textual datasets without excessive memory usage

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of implementing a course recommendation system using machine learning presents a promising outlook. Initially, there will be upfront costs associated with acquiring and integrating the necessary technology, employing skilled data scientists, and developing the algorithm. However, these initial investments are likely to yield substantial long-term benefits. The implementation of a robust recommendation system can enhance student engagement and satisfaction, leading to increased retention rates and ultimately higher tuition revenues. Additionally, the system's ability to optimize course selection can contribute to improved academic performance, potentially reducing dropout rates and associated financial losses for both students and institutions.

Furthermore, the automation of the recommendation process can lead to operational efficiencies, freeing up administrative resources for more strategic tasks. As the system continuously learns and adapts to evolving academic trends, it ensures that the institution remains competitive in attracting and retaining students.

3.3.2 Technical Feasibility

The technical feasibility of a course recommendation system project using machine learning is high, leveraging the advancements in artificial intelligence and data analytics. Machine learning algorithms can effectively analyze large datasets of user preferences, historical course selections, and academic performance to generate personalized recommendations. The system can employ collaborative filtering, contentbased filtering, or hybrid models to enhance accuracy. With the availability of diverse open-source libraries like TensorFlow and scikit-learn, developers can easily implement and fine-tune machine learning models. Additionally, cloud computing platforms such as AWS or Azure can handle the computational demands for training and deploying these models at scale.

Integration with educational databases, learning management systems, and user profiles can be achieved through standardized APIs or custom connectors. The feasibility is further augmented by the increasing accessibility of educational data and the willingness of institutions to adopt smart technologies. However, potential challenges may include data privacy concerns, ensuring algorithmic fairness, and the need for continuous model optimization to adapt to evolving educational trends.

3.3.3 Social Feasibility

The social feasibility of a course recommendation system project utilizing machine learning is promising and aligned with contemporary educational trends. Such a system addresses the evolving needs of students, offering personalized guidance in navigating the vast landscape of available courses. By leveraging machine learning algorithms, the system can analyze individual learning preferences, academic strengths, and career aspirations, ensuring tailored recommendations that resonate with students on a personal level. This not only enhances the overall learning experience but also fosters a sense of empowerment and engagement among students, as they feel more in control of their educational journey.

Furthermore, the project aligns with the broader societal shift towards technology-driven solutions in education, reflecting the growing acceptance and integration of artificial intelligence in various aspects of our lives. However, it is crucial to consider potential concerns related to data privacy, algorithmic bias, and equitable access to technology, ensuring that the system is ethically designed and benefits all students irrespective of their socio-economic background.

3.4 System Specification

3.4.1 Hardware Specification

• A system with a multi-core processor such as intel i5, or AMD Ryzen5 or higher would be useful for dealing with large datasets.
• GPU could prove beneficial in vectorizing the data which includes finding the frequency of the words in the document which is a huge task. Preferred GPU Nvidia GeForce RTX series.
• A computer with a RAM of 8GB and containing SSD for faster data retrieval.

3.4.2 Software Specification

• IDE: Visual Studio Code
• ML Libraries: scikit-learn,NearestNeighbors,TF-IDF Vectorizer
• Web-Development Libraries: flask, CORS
• UI Framework: Flutter, Dart language

3.4.3 Standards and Policies

Flutter
Flutter is an open-source UI (User Interface) toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It uses the Dart programming language and follows a reactive programming paradigm.
Standard Used: ISO 25010

Python
Python stands out in the realm of machine learning due to its simplicity, readability, and expansive library ecosystem tailored for ML tasks. With frameworks like TensorFlow, PyTorch, Scikit-learn, and Keras, Python offers a robust toolkit for data manipulation, model building, and evaluation, streamlining the entire ML workflow.
Standard Used: PEP 257

Chapter 4

METHODOLOGY

4.1 Course Recommender Architecture

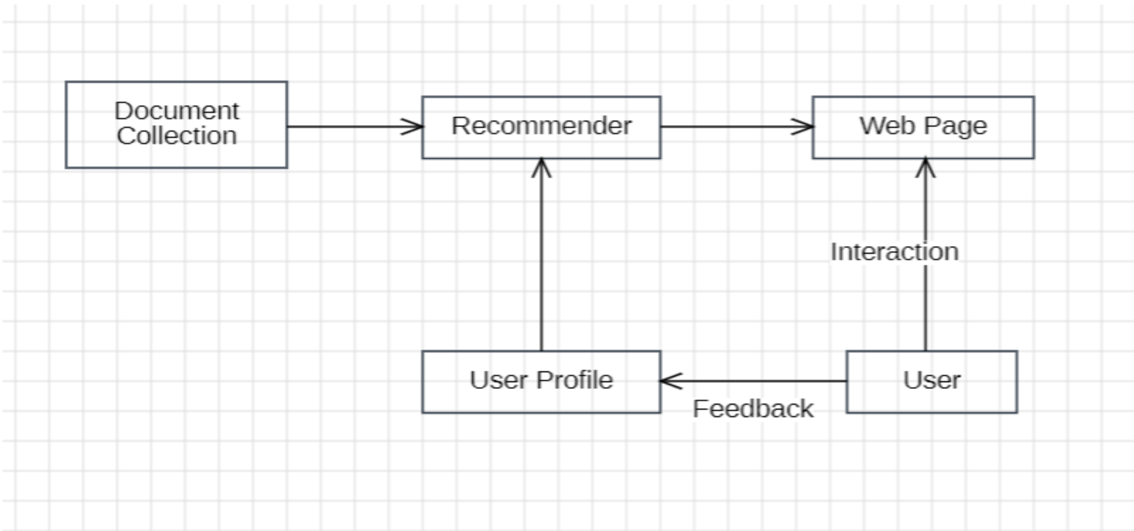


Figure 4.1: **Architecture Diagram for Course Recommendation System**

Figure 4.1 depicts the architecture diagram for the course recommendation system project comprises several key components. The document collection module gathers information on available courses, while the recommender utilizes this data to generate personalized recommendations based on user preferences.The web page component serves as the user interface, presenting recommended courses and enabling user interactions. User profiles store individual preferences and feedback, facilitating a dynamic and personalized learning experience within the course recommendation system.

4.2 Design Phase

4.2.1 Data Flow Diagram

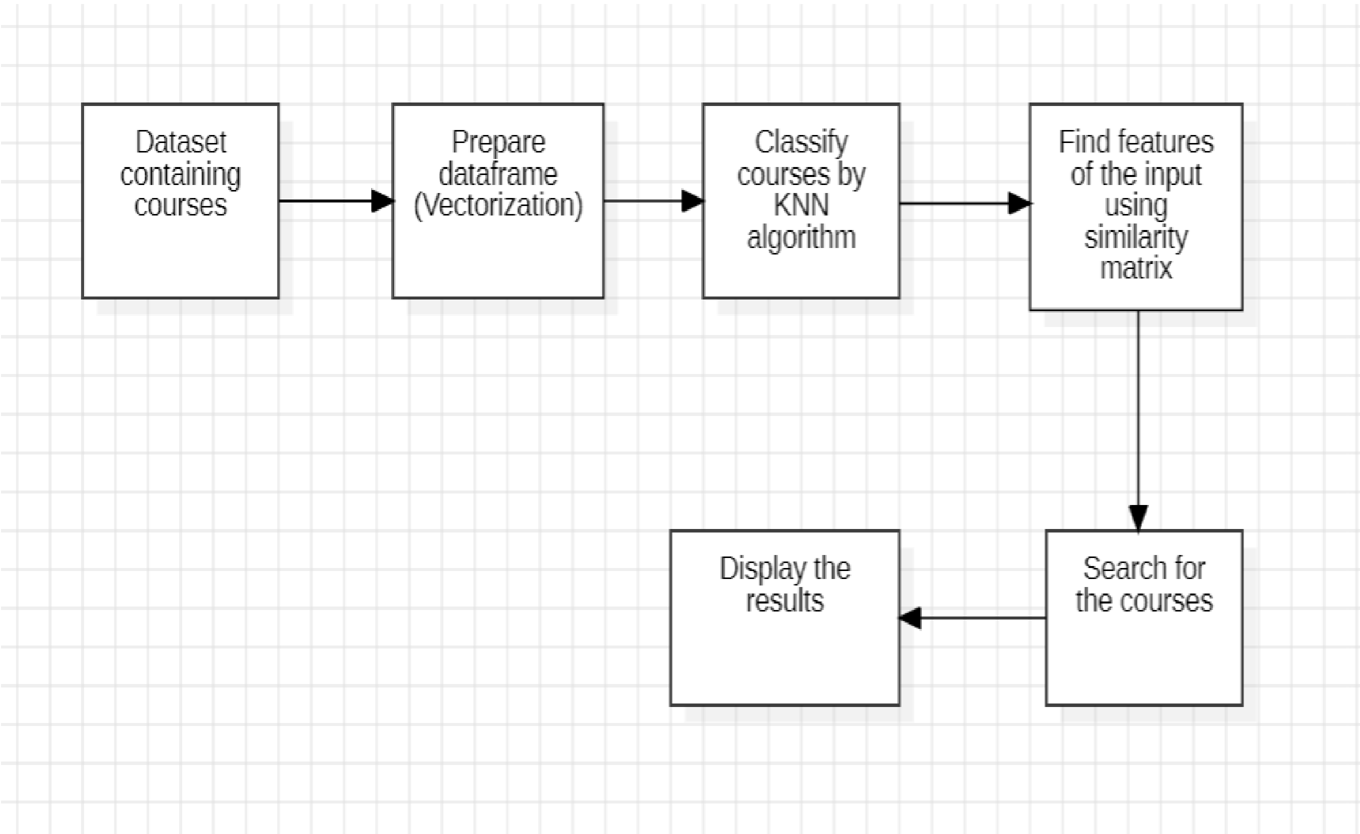


Figure 4.2: **Data Flow Diagram for Course Recommendation System**

Figure 4.2 demonstrates the data flow diagram in which the first step is to collect data about the user, such as their past course choices, grades, and interests.Next, the system collects data about the courses available, such as the course content, instructor information, and student reviews. Once the data has been collected, the system uses a machine learning algorithm to analyze it and identify patterns. The data flow diagram also shows that the system can track the user's progress in the recommended courses. This information can be used to further refine the recommendations and provide the user with even more relevant suggestions in the future.Overall, the flowchart provides a good overview of how a course recommendation system powered by machine learning can work. By analyzing data about users and courses, the system can generate personalized recommendations that help users find the courses that are right for them.

4.2.2 Use Case Diagram

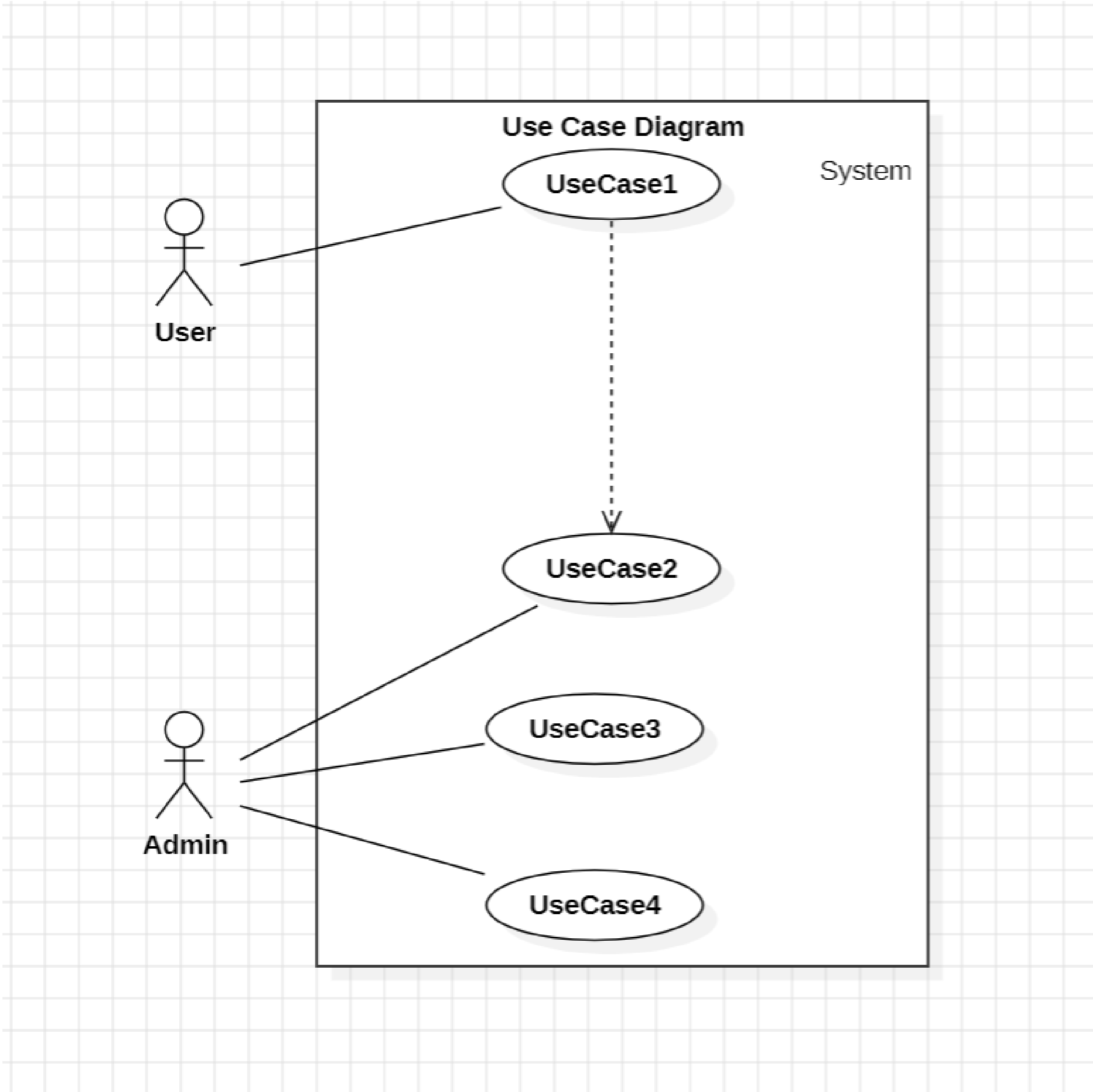


Figure 4.3: **Use Case Diagram for Course Recommendation System**

Figure 4.3 portrays the use case diagram where the user can provide the system with data in a few different ways. They can provide a dataset of their past course choices, or they can update the system with new courses that they are interested in. The system can also be integrated with a web service, which would allow the user to get recommendations from the system without having to provide any data themselves. Once the system has data about the user, it can use that data to generate recommendations. The diagram shows that the system can recommend courses based on the user's past course choices, the language of the course, and whether or not the course is paid.

4.2.3 Class Diagram

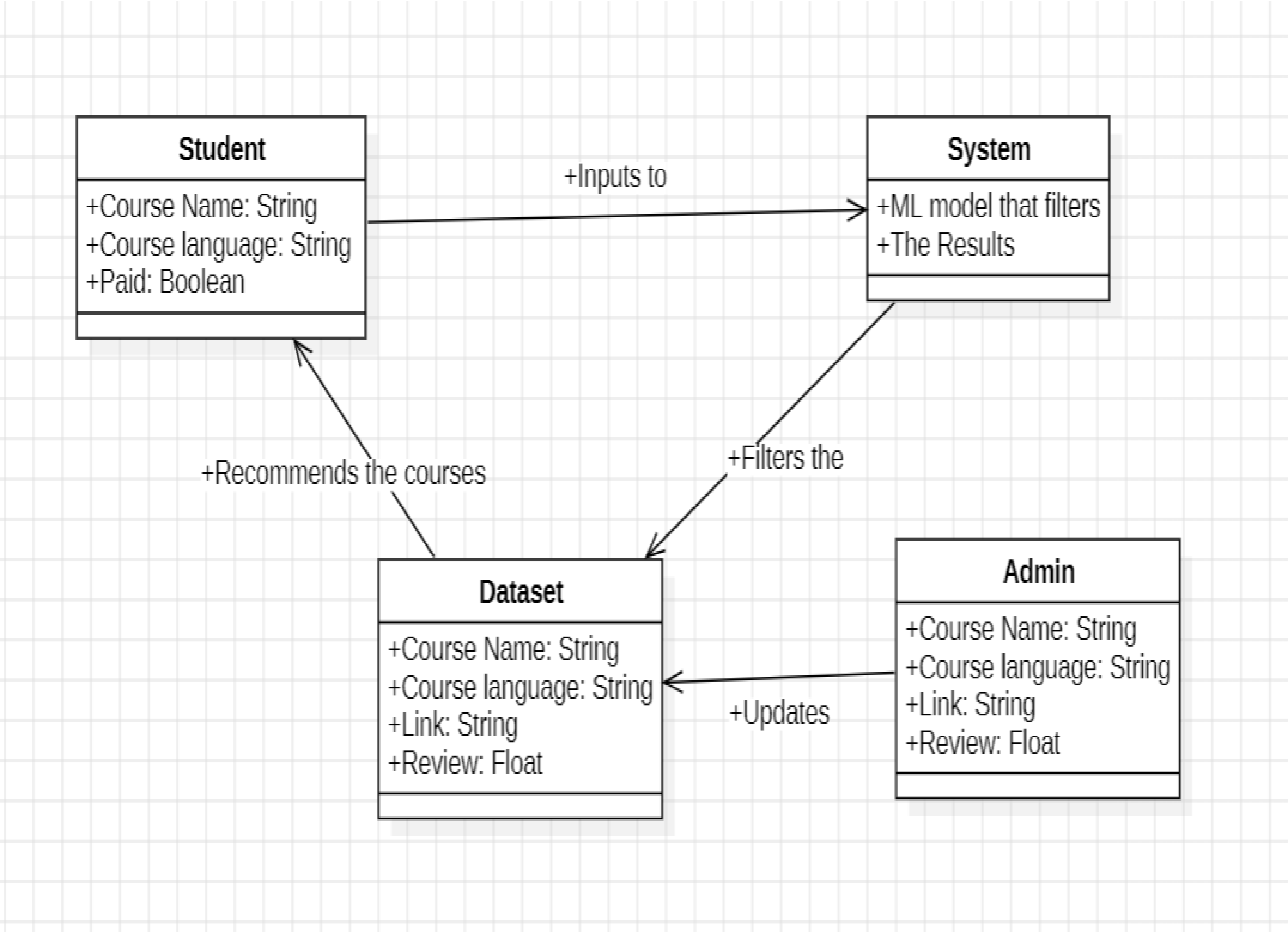


Figure 4.4: **Class Diagram for Course Recommendation System**

Figure 4.4 demonstrates the class diagram of the course recommender system.The system is made up of three main parts: the student, the inputs to the system, and the system itself. The student can input their course name, course language, and whether or not the course is paid. The system then uses this information to filter a dataset of courses and recommend relevant ones to the student. The system can also be updated by the admin, who can add or remove courses from the dataset.

4.2.4 Sequence Diagram

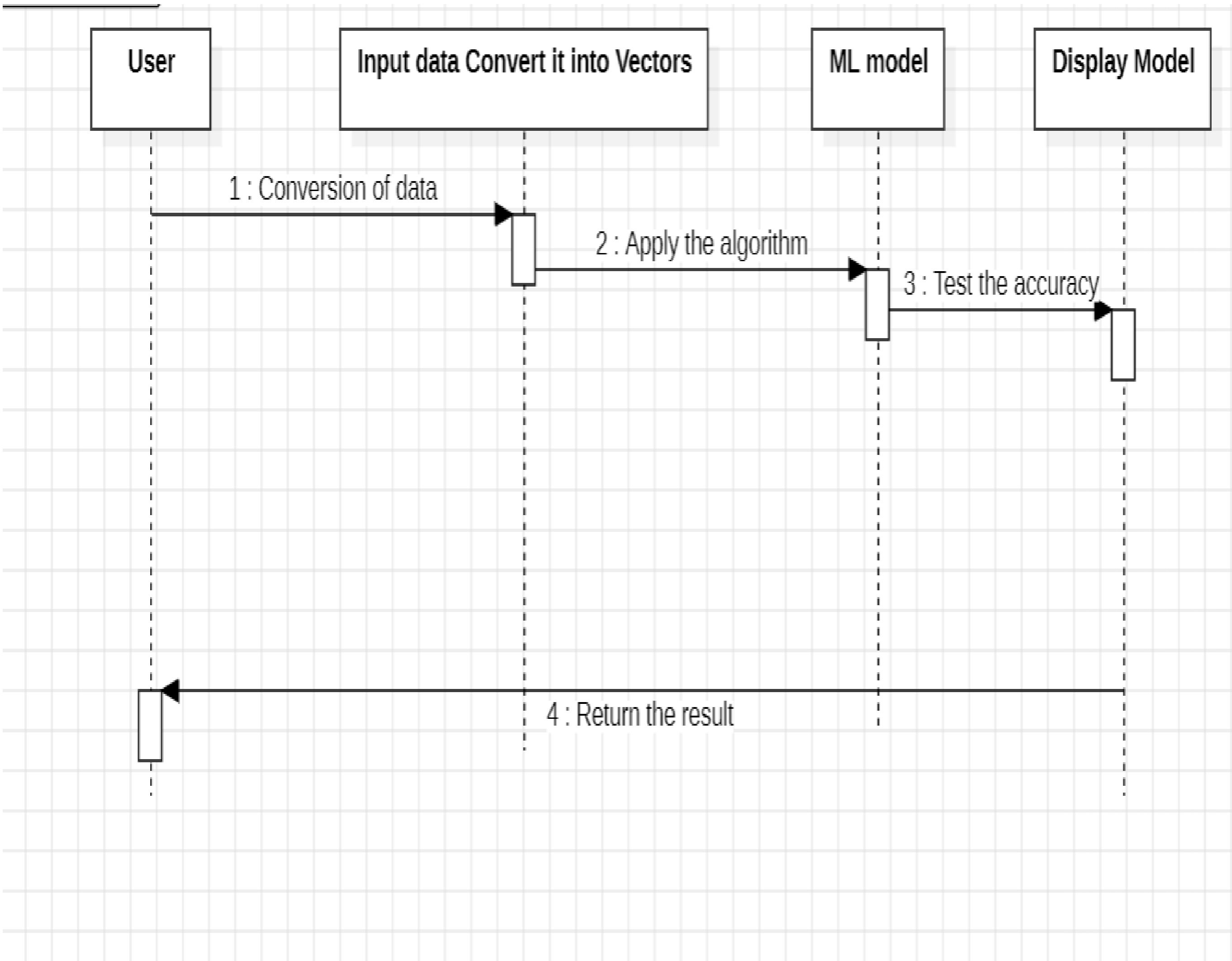


Figure 4.5: **Sequence Diagram for Course Recommendation System**

Figure 4.5 demonstrates the sequence of operations in Course Recommender System. Data is preprocessed, which may involve scaling, cleaning, and transforming the data into a format suitable for the algorithm. Then, the data is split into two sets: a training set and a test set. The training set is used to train the model.The model is evaluated using the test set. This involves making predictions on the test data and comparing them to the actual values.The accuracy of the model is calculated. If the accuracy is not satisfactory, the process may be repeated with different parameters or a different machine learning algorithm.Finally, the model is used to make predictions on new data.

4.2.5 Collaboration diagram

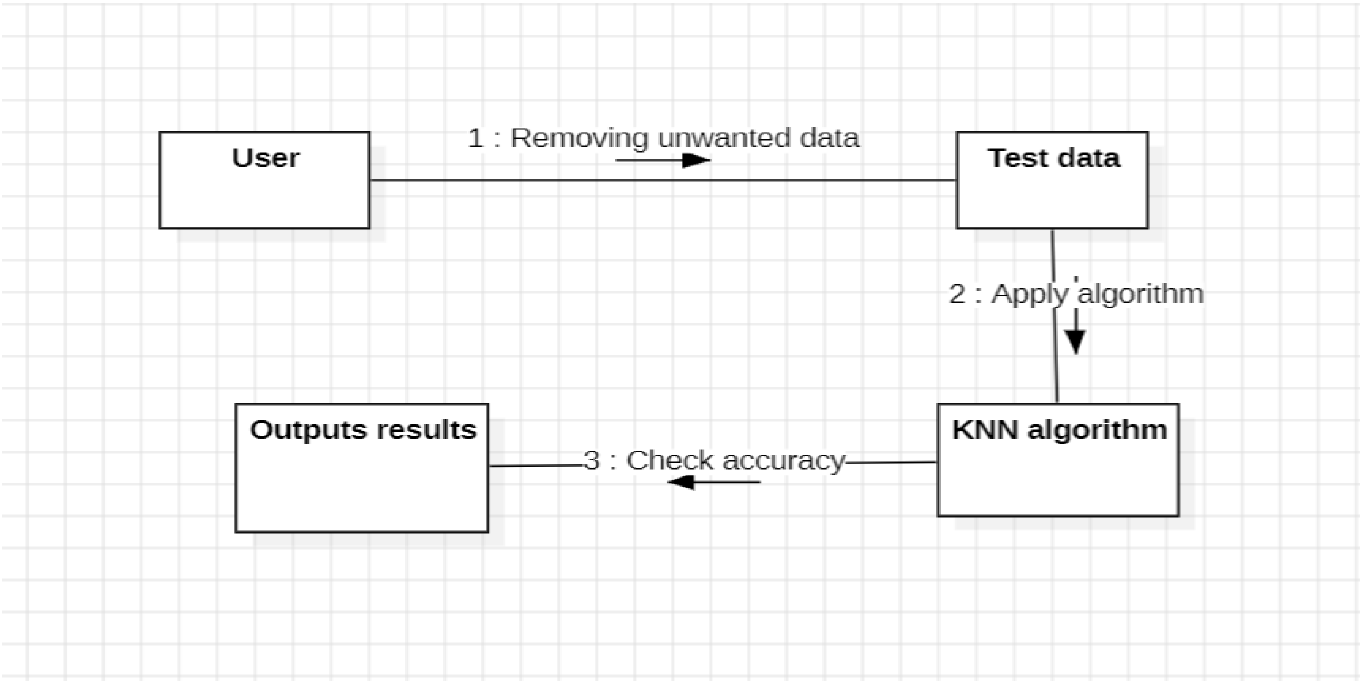


Figure 4.6: **Collaboration Diagram for Course Recommendation System**

Figure 4.6 gives us an overview of the collaboration diagram of a course recommender system. It depicts a machine learning model's prediction process, potentially using nearest neighbors algorithm. Data is first preprocessed to make it suitable for the chosen algorithm. Next, it's split into training and test sets. The training set trains the model, while the test set evaluates its performance.If the model's accuracy meets expectations, it's ready to make predictions on new data. Otherwise, the parameters or algorithm can be tweaked and the process repeated.

4.2.6 Activity Diagram

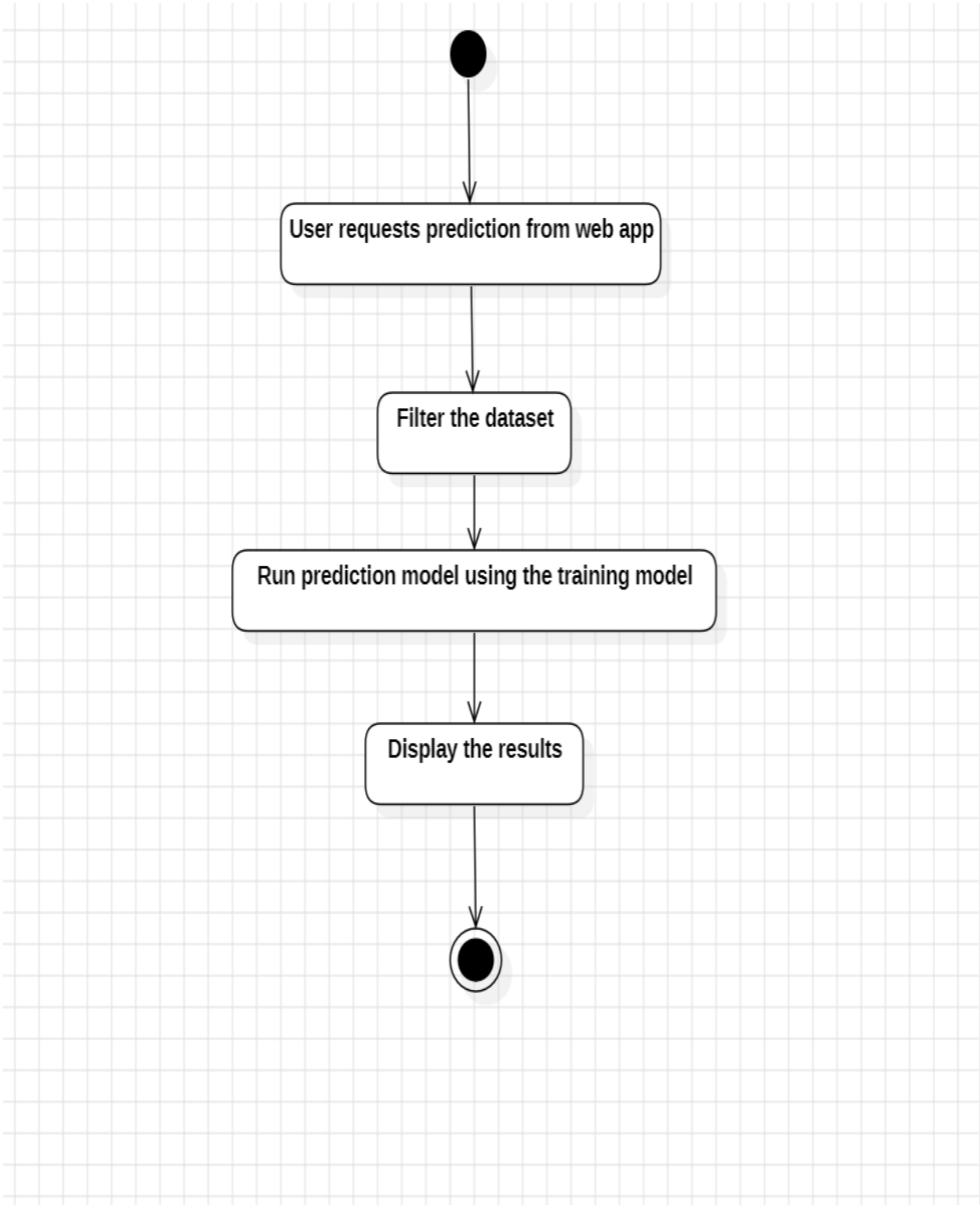


Figure 4.7: **Activity Diagram for Course Recommendation System**

Figure 4.7 illustrates the activity diagram of the Course Recommender System. The user submits a request, which is then filtered by the app's dataset. The filtered data is then used to run a prediction model, trained on existing data, and the results are displayed to the user. This cyclical process allows the app to learn and refine its predictions over time.

4.3 Algorithm & Pseudo Code

4.3.1 Vectorization with TF-IDF and Classification using K-Nearest Neighbors

Vectorization using TF-IDF:

Steps:

1. Term Frequency (TF): Calculate the frequency of each term in a document.

2. Inverse Document Frequency (IDF): Determine the importance of a term across all documents in the corpus.

3. TF-IDF Calculation: : Compute TF-IDF values for each term-document pair.

K-Nearest Neighbors Algorithm:

Steps:

1. Selecting k-Nearest Neighbors: : Select the k-nearest neighbors based on the smallest distances.

2. Majority Voting (for Classification): For classification, determine the majority class among the k-nearest neighbors to predict the class of the new data point.

4.3.2 Pseudo Code

Vectorization using TF-IDF:

```
function TF IDF_Vectorization(corpus) :
      for each document in corpus:
          tokenize document into individualterms
          calculate term frequency (TF) for each term in document
     calculate inverse document frequency (IDF) for each term in corpus
     calculate TF IDF for each term document pair
     normalize TF IDF values
     return TF IDF matrix
```

K-Nearest Neighbors Algorithm:

```
function kNearest Neighbors (data,newpoint,k):
for each data point in dataset:
   calculate distance between newpoint and data point
sort distances and select top k nearest neighbors
for classification:
```

```
    determine majority class among k nearest neighbors
    return predicted class
for regression:
    calculate average of values among knearest neighbors
    return predicted value
```

4.4 Module Description

4.4.1 Module 1: Data collection and Processing

• **Course Data Source:** Obtain course information from diverse sources like educational platforms (Coursera, Udemy, etc.), university databases, or curated datasets available online.

• **Data Cleaning:** Address inconsistencies, remove duplicates, handle missing values, and format data for uniformity.

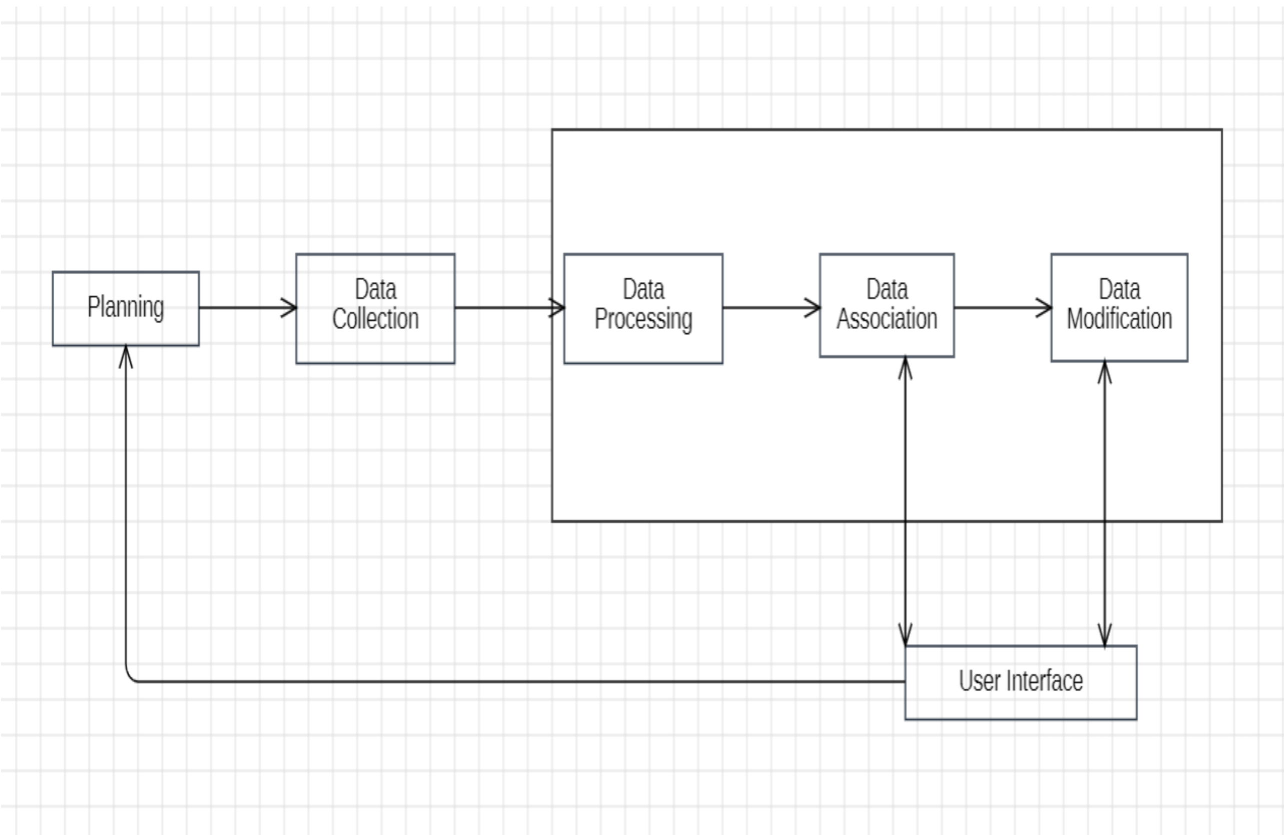


Figure 4.8: **Architecture Diagram for Data Collection and Data Processing**

4.4.2 Module 2: Vectorization (TF IDF vectorizer)

• Vectorization is the process of converting textual data into numerical vectors and is a process that is usually applied once the text is cleaned. It can help improve the execution speed and reduce the training time of your code. In this article, we will discuss some of the best techniques to perform vectorization.
• The TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is a common tool used in machine learning (ML) to convert textual data into a matrix of TF-IDF features.
• The break down of wha TF-IDF does:
• **Term Frequency (TF)** measures the frequency of a term (word) in a document. It's calculated as the number of times a term appears in a document divided by the total number of terms in the document.
• **Inverse Document Frequency (IDF)** measures how important a term is across a collection of documents. Terms that occur frequently across many documents are considered less important. IDF is calculated as the logarithm of the total number of documents divided by the number of documents containing the term.

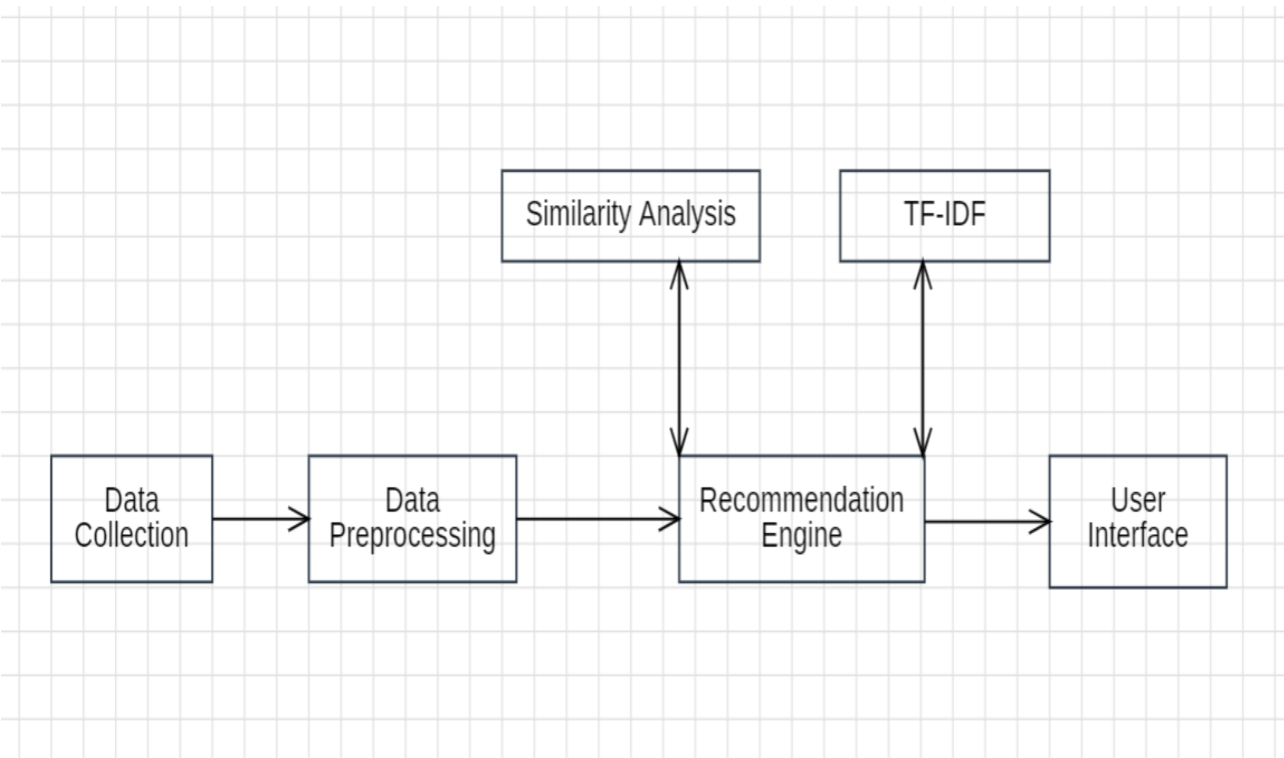


Figure 4.9: **Architecture Diagram for Vectorization Module**

4.4.3 Module 3: KNN Algorithm

• The K-Nearest Neighbor (KNN) algorithm is a popular machine learning technique used for classification and regression tasks. It relies on the idea that similar data points tend to have similar labels or values.
• KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
• During the training phase, the KNN algorithm stores the entire training dataset as a reference. When making predictions, it calculates the distance between the input data point and all the training examples, using a chosen distance metric such as Euclidean distance.
• KNN is a reasonably simple classification technique that identifies the class in which a sample belongs by measuring its similarity with other nearby points.Though it is elementary to understand, it is a powerful technique for identifying the class of an unknown sample point.

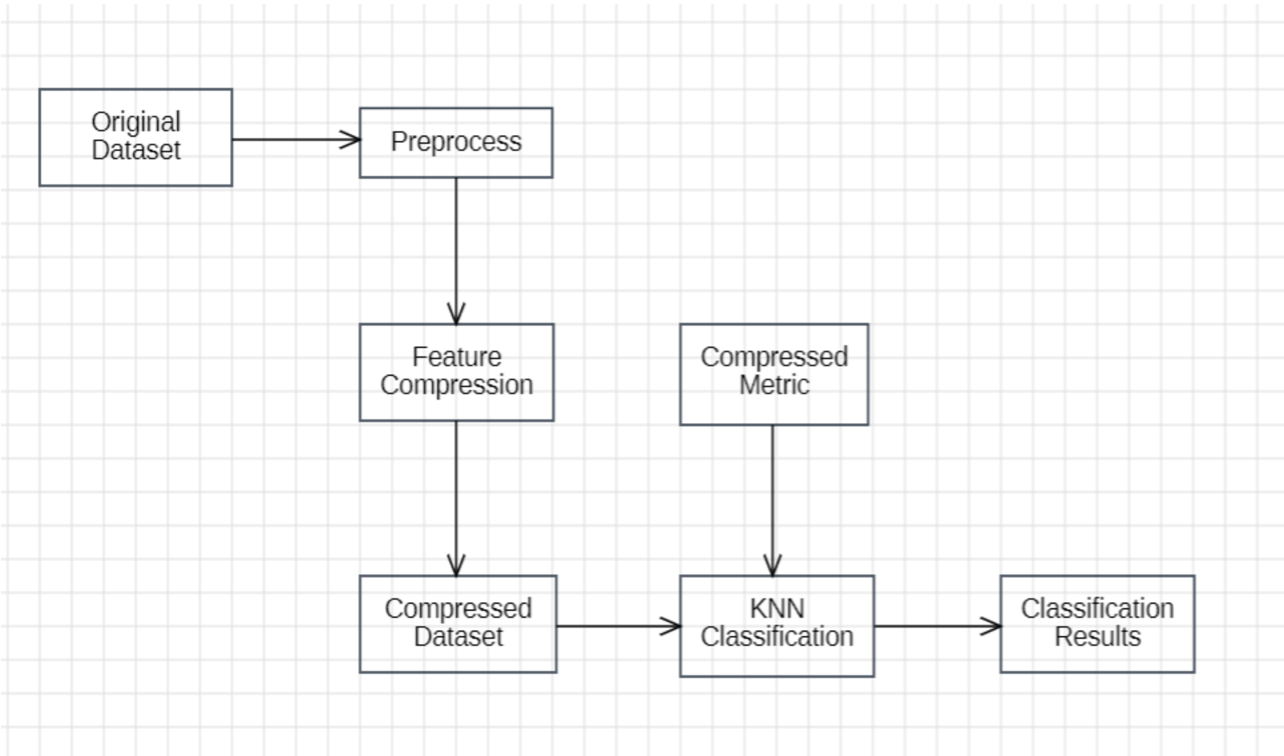


Figure 4.10: **Architecture Diagram for KNN Algorithm**

4.5 Steps to implement the project

4.5.1 Step 1: Data Collection and Processing

• The data was collected through various datasets that are available online which included course name, course description, paid, difficulty of the course, rating of the course and the url to the course.

• Once the datasets are obtained all the data is merged into one sheet and removed unwanted columns.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	course_id	course_tit	url	is_paid	price	num_subscribers	num_reviews	num_lectures	level	content_duration	published_timestamp	subject	profit	published_	published_	year	month	day
2	1070968	Ultimate I	https://wv	TRUE	200	2147	23	51	All Levels	1.5 hours	2017-01-18T20:58:58Z	Business Finance	429400	########	20:58:58Z	2017	1	18
3	1113822	Complete	https://wv	TRUE	75	2792	923	274	All Levels	39 hours	2017-03-09T16:34:20Z	Business Finance	209400	########	16:34:20Z	2017	3	9
4	1006314	Financial M	https://wv	TRUE	45	2174	74	51	Intermediate Leve	2.5 hours	2016-12-19T19:26:30Z	Business Finance	97830	########	19:26:30Z	2016	12	19
5	1210588	Beginner t	https://wv	TRUE	95	2451	11	36	All Levels	3 hours	2017-05-30T20:07:24Z	Business Finance	232845	########	20:07:24Z	2017	5	30
6	1011058	How To M	https://wv	TRUE	200	1276	45	26	Intermediate Leve	2 hours	2016-12-13T14:57:18Z	Business Finance	255200	########	14:57:18Z	2016	12	13
7	192870	Trading Pe	https://wv	TRUE	150	9221	138	25	All Levels	3 hours	2014-05-02T15:13:30Z	Business Finance	1383150	########	15:13:30Z	2014	5	2
8	739964	Investing A	https://wv	TRUE	65	1540	178	26	Beginner Level	1 hour	2016-02-21T18:23:12Z	Business Finance	100100	########	18:23:12Z	2016	2	21
9	403100	Trading St	https://wv	TRUE	95	2917	148	23	All Levels	2.5 hours	2015-01-30T22:13:03Z	Business Finance	277115	########	22:13:03Z	2015	1	30
10	476268	Options Tr	https://wv	TRUE	195	5172	34	38	Expert Level	2.5 hours	2015-05-28T00:14:03Z	Business Finance	1008540	########	00:14:03Z	2015	5	28
11	1167710	The Only I	https://wv	TRUE	200	827	14	15	All Levels	1 hour	2017-04-18T18:13:32Z	Business Finance	165400	########	18:13:32Z	2017	4	18
12	592338	Forex Trad	https://wv	TRUE	200	4284	93	76	All Levels	5 hours	2015-09-11T16:47:02Z	Business Finance	856800	########	16:47:02Z	2015	9	11
13	975046	Trading Op	https://wv	TRUE	200	1380	42	17	All Levels	1 hour	2016-10-18T22:52:31Z	Business Finance	276000	########	22:52:31Z	2016	10	18
14	742602	Financial M	https://wv	TRUE	30	3607	21	19	All Levels	1.5 hours	2016-02-03T18:04:01Z	Business Finance	108210	########	18:04:01Z	2016	2	3
15	794151	Forex Trad	https://wv	TRUE	195	4061	52	16	All Levels	2 hours	2016-03-16T15:40:19Z	Business Finance	791895	########	15:40:19Z	2016	3	16
16	1196544	Python Alg	https://wv	TRUE	200	294	19	42	All Levels	7 hours	2017-04-28T16:41:44Z	Business Finance	58800	########	16:41:44Z	2017	4	28
17	504036	Short Sellir	https://wv	TRUE	75	2276	106	19	Intermediate Leve	1.5 hours	2015-06-22T21:18:35Z	Business Finance	170700	########	21:18:35Z	2015	6	22
18	719698	Basic Tech	https://wv	TRUE	20	4919	79	16	Beginner Level	1.5 hours	2016-01-08T17:21:26Z	Business Finance	98380	########	17:21:26Z	2016	1	8
19	564966	The Compl	https://wv	TRUE	200	2666	115	52	All Levels	4 hours	2015-08-10T21:07:35Z	Business Finance	533200	########	21:07:35Z	2015	8	10
20	606928	7 Deadly M	https://wv	TRUE	50	5354	24	23	All Levels	1.5 hours	2015-09-21T18:10:34Z	Business Finance	267700	########	18:10:34Z	2015	9	21
21	58977	Financial S	https://wv	TRUE	95	8095	249	12	Beginner Level	35 mins	2013-06-09T00:21:26Z	Business Finance	769025	########	00:21:26Z	2013	6	9
22	1242604	Winning Fc	https://wv	TRUE	200	809	3	25	All Levels	2 hours	2017-06-06T02:54:04Z	Business Finance	161800	########	02:54:04Z	2017	6	6
23	798740	Forex Trad	https://wv	TRUE	200	2295	84	39	All Levels	4 hours	2016-05-02T19:26:48Z	Business Finance	459000	########	19:26:48Z	2016	5	2
24	506568	Create A B	https://wv	TRUE	75	10149	83	16	All Levels	2 hours	2015-05-26T17:25:46Z	Business Finance	761175	########	17:25:46Z	2015	5	26
25	1020760	Introductic	https://wv	TRUE	50	1916	38	23	Beginner Level	1 hour	2016-12-05T22:14:17Z	Business Finance	95800	########	22:14:17Z	2016	12	5
26	859742	The Compl	https://wv	TRUE	45	2507	333	40	All Levels	2 hours	2016-05-30T18:53:46Z	Business Finance	112815	########	18:53:46Z	2016	5	30
27	383414	Beginner t	https://wv	TRUE	150	5786	518	50	All Levels	5 hours	2015-01-31T15:34:05Z	Business Finance	867900	########	15:34:05Z	2015	1	31
28	353044	Investing 1	https://wv	TRUE	200	5334	373	42	All Levels	4.5 hours	2014-11-28T17:49:03Z	Business Finance	1066800	########	17:49:03Z	2014	11	28
29	447362	Create You	https://wv	TRUE	175	4005	237	25	Intermediate Leve	2 hours	2015-04-12T20:13:47Z	Business Finance	700875	########	20:13:47Z	2015	4	12

Figure 4.11: **Raw data**

	A	B	C	D	E
1	Course Name	Difficulty L	Course Rating	Course URL	Course Description
2	Write A Feature Length Screenplay For Film Or Television	Beginner	4.8	https://www.coursera.org/learn/wr	Write a Full Length Feature Film Script In this course, you will write a complete, feature-length scre
3	Business Strategy: Business Model Canvas Analysis with Miro	Beginner	4.8	https://www.coursera.org/learn/ca	By the end of this guided project, you will be fluent in identifying and creating Business Model Canv
4	Silicon Thin Film Solar Cells	Advanced	4.1	https://www.coursera.org/learn/sili	This course consists of a general presentation of solar cells based on silicon thin films. It is the thi
5	Finance for Managers	Intermedia	4.8	https://www.coursera.org/learn/op	When it comes to numbers, there is always more than meets the eye. In operational finance, you
6	Retrieve Data using Single-Table SQL Queries	Beginner	4.6	https://www.coursera.org/learn/sin	In this course youï¿½ll learn how to effectively retrieve data from a relational database table using
7	Building Test Automation Framework using Selenium and TestNG	Beginner	4.7	https://www.coursera.org/learn/bu	Selenium is one of the most widely used functional UI automation testing tools and TestNG is a bri
8	Doing Business in China Capstone	Advanced	3.3	https://www.coursera.org/learn/do	Doing Business in China Capstone enables you to apply your skills to real business challenges. Youï
9	Programming Languages, Part A	Intermedia	4.9	https://www.coursera.org/learn/pr	This course is an introduction to the basic concepts of programming languages, with a strong emph
10	The Roles and Responsibilities of Nonprofit Boards of Directors within th	Intermedia	4.3	https://www.coursera.org/learn/no	This course provides a more in-depth look at the roles and responsibilities of boards of directors in
11	Business Russian Communication. Part 3	Intermedia	Not Calibrated	https://www.coursera.org/learn/bu	Russian is considered to be one of the most difficult languages in the world, and many learners stru
12	Agile Projects: Developing Tasks with Taiga	Beginner	4	https://www.coursera.org/learn/de	By the end of this guided project, you will be fluent in creating tasks for Agile projects based on pre
13	Esports Management Capstone Project	Beginner	4.4	https://www.coursera.org/learn/es	We will analyze the pros and cons of creating a single or multiple Esport organization and recomme
14	Hacking and Patching	Advanced	3.4	https://www.coursera.org/learn/ha	In this MOOC, you will learn how to hack web apps with command injection vulnerabilities in a we
15	Business Statistics and Analysis Capstone	Beginner	4.7	https://www.coursera.org/learn/bu	The Business Statistics and Analysis Capstone is an opportunity to apply various skills developed ac
16	Grab Data Fast with Vertical and Horizontal LOOKUP	Beginner	4.5	https://www.coursera.org/learn/gr	Data can come our way in multiple forms and from multiple file types. Itï¿½s likely that at some po
17	Global Health: An Interdisciplinary Overview	Beginner	4.6	https://www.coursera.org/learn/glc	This course proposes an overview of current global health challenges drawing on the insights of se
18	Python Programming Essentials	Beginner	4.8	https://www.coursera.org/learn/py	This course will introduce you to the wonderful world of Python programming! We'll learn about t
19	Creating Dashboards and Storytelling with Tableau	Advanced	4.6	https://www.coursera.org/learn/da	Leveraging the visualizations you created in the previous course, Visual Analytics with Tableau, you
20	Parallel programming	Beginner	4.4	https://www.coursera.org/learn/pa	With every smartphone and computer now boasting multiple processors, the use of functional idea
21	Recommendation Systems with TensorFlow on GCP	Advanced	4.2	https://www.coursera.org/learn/re	In this course, you'll apply your knowledge of classification models and embeddings to build a ML p
22	The Changing Arctic	Not Calibr	4.3	https://www.coursera.org/learn/ch	What will I learn? After taking this course you will have an understanding of Arctic landscapes, how
23	COVID-19 - A clinical update	Beginner	4.7	https://www.coursera.org/learn/co	As an expert in infectious diseases, editor of the Journal of Infectious Diseases and author of the t
24	How to Create Text Effects in GIMP	Beginner	4.7	https://www.coursera.org/learn/ho	In this 1-hour long project-based course, you will learn how to add a myriad of text effects. You wi
25	Preparing for the Google Cloud Professional Data Engineer Exam	Beginner	4.5	https://www.coursera.org/learn/pr	From the course: "The best way to prepare for the exam is to be competent in the skills required o
26	Multiple Regression Analysis in Public Health	Intermedia	4.6	https://www.coursera.org/learn/mu	Biostatistics is the application of statistical reasoning to the life sciences, and it's the key to unlock
27	General Pathophysiology	Intermedia	4	https://www.coursera.org/learn/pa	Dear listeners! Warning: this course contains shocking materials and is not recommended for view
28	Population Health: Alternative Payment Models	Advanced	4.9	https://www.coursera.org/learn/alt	The way we currently pay our care providers lead to suboptimal outcomes and unnecessary spendi
29	AWS Elastic Beanstalk: Build & Deploy a Node.js RESTful API	Advanced	5	https://www.coursera.org/learn/no	In this 1-hour long project-based course, you will learn how to create a Node.js RESTful API & laun
30	Philosophy, Science and Religion: Philosophy and Religion	Intermedia	4.6	https://www.coursera.org/learn/ph	Philosophy, Science and Religion mark three of the most fundamental modes of thinking about the

Figure 4.12: **Processed data**

4.5.2 Step 2: Building the model and the back end using flask

• Install all python dependencies that are required for building the model using the "pip install [package-name]" command in the terminal.
• Vectorize the data and build the knn model using the vectors and set the nneighbours to 100 so that it gives us the 100 relevant results.
• Once this is all done in the server route /predict define a predict function that takes in the input form the UI and stores in a variable user-input and predicts the results based on the user input.

4.5.3 Step 3: UI designing using Flutter

• The UI is designed using the Flutter framework in dart language. The following are the components in the UI:
• An input field with a search icon.
• A widget that updates dynamically to show the recommended courses.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design: Course Recommendation System

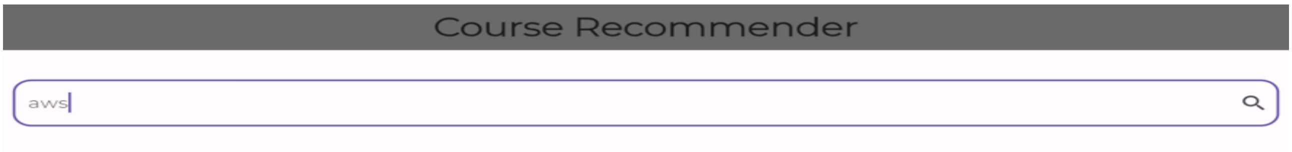


Figure 5.1: **Input design of Course Recommender**

Figure 5.1 portrays the input design of the application include an input field to search the courses.On clicking the search icon the UI sends a request to the server to recommend the courses.

5.1.2 Output Design: Course Recommendation System

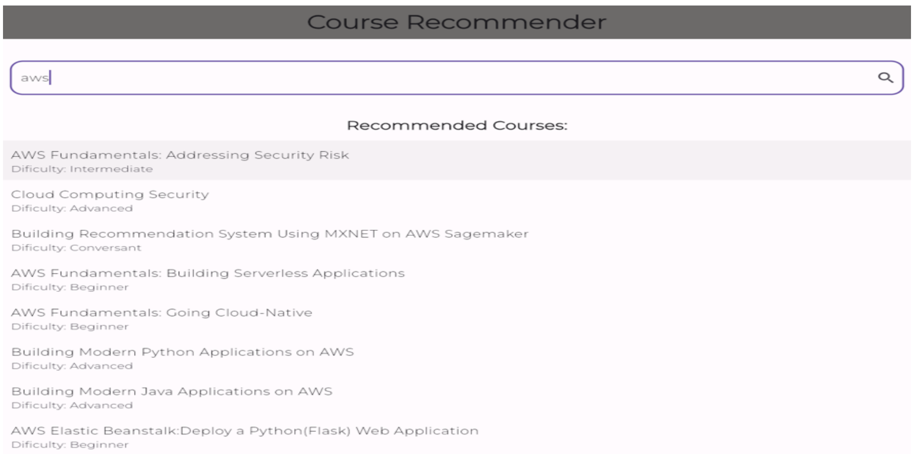


Figure 5.2: **Output design of Course Recommender**

Figure 5.2 portrays the output is shown in the same scaffold the home screen. In Flutter we use an expanded class to accomodate the vast results so as to make the screen scrollable.

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Testing the TF-IDF vectorizer unit.

Input

```
# Vectorize the course descriptions
tfidf = TfidfVectorizer(stop_words='english')
course_vectors = tfidf.fit_transform(courses_df['Course Description'])
print(course_vectors)
```

Figure 5.3: **Unit Test Input**

The unit testing involves the testing of all the units or components in the system. For us the module using the vectorization is a unit so we are testing the vectorization module by giving an input of the course data and print the result.

Test result

```
 * Restarting with watchdog (windowsapi)
  (0, 13157)    0.05942005434392607
  (0, 16980)    0.068966442172866
  (0, 9868)     0.05984401631322437
  (0, 14610)    0.07189021655404564
  (0, 13963)    0.04306006743038402
  (0, 14357)    0.0547364460432454
  (0, 4374)     0.03583346412910152
  (0, 13031)    0.053192486044520364
  (0, 14606)    0.0508971418116341
  (0, 15281)    0.04946943099382932
  (0, 3852)     0.055233123890227005
  (0, 7532)     0.06678264628406667
  (0, 19644)    0.024331639701590444
  (0, 4127)     0.048475262627742606
  (0, 17287)    0.04603436228337483
  (0, 9328)     0.03888982081542526
  (0, 5742)     0.06503891894317988
  (0, 7266)     0.03873533680574497
  (0, 14736)    0.05260645525932396
  (0, 19008)    0.07632903411300661
  (0, 2984)     0.07632903411300661
  (0, 16971)    0.03522481838191546
  (0, 16254)    0.07382922170112527
  (0, 7643)     0.036607010516929564
  (0, 5726)     0.05360634981782513
  :     :
  (3521, 14639) 0.06587452080642048
  (3521, 9771)  0.06688604761417578
  (3521, 15593) 0.147635720885459
  (3521, 13322) 0.1329768458240359
  (3521, 2609)  0.06113076491163066
  (3521, 14129) 0.057007059379253044
  (3521, 8409)  0.05684712220930083
  (3521, 10875) 0.11484458814258167
  (3521, 16504) 0.056377847283058447
  (3521, 1316)  0.05889669732605204
  (3521, 16492) 0.06841745306545047
  (3521, 19659) 0.08552378842980934
  (3521, 4740)  0.07807762715346465
  (3521, 11098) 0.10814971266280128
  (3521, 18311) 0.07075603393002998
```

Figure 5.4: **Unit Test Output**

The output would be a matrix consisting of the frequency of words in rows and columns.

5.3.2 Integration testing

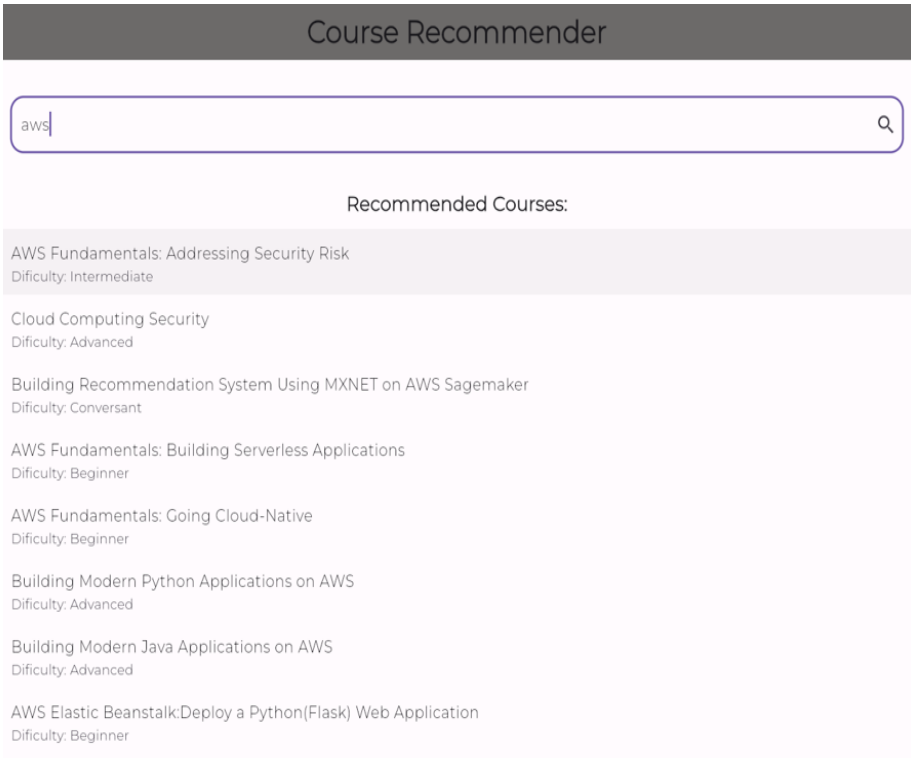


Figure 5.5: **Integration Test Input**

Integration testing for our project includes the testing of the server endpoints to check if the server is up and running and accepting the requests generarted from the UI.The UI is also responsive and the result is shown.

Test result

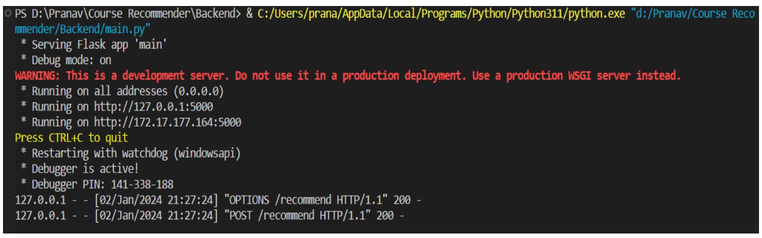


Figure 5.6: **Integration Test Output**

In the ouput we can see the server is up and running and also accepting requests form the user and it can be seen in the server logs which include the users IP address and path of the request.

5.3.3 System testing

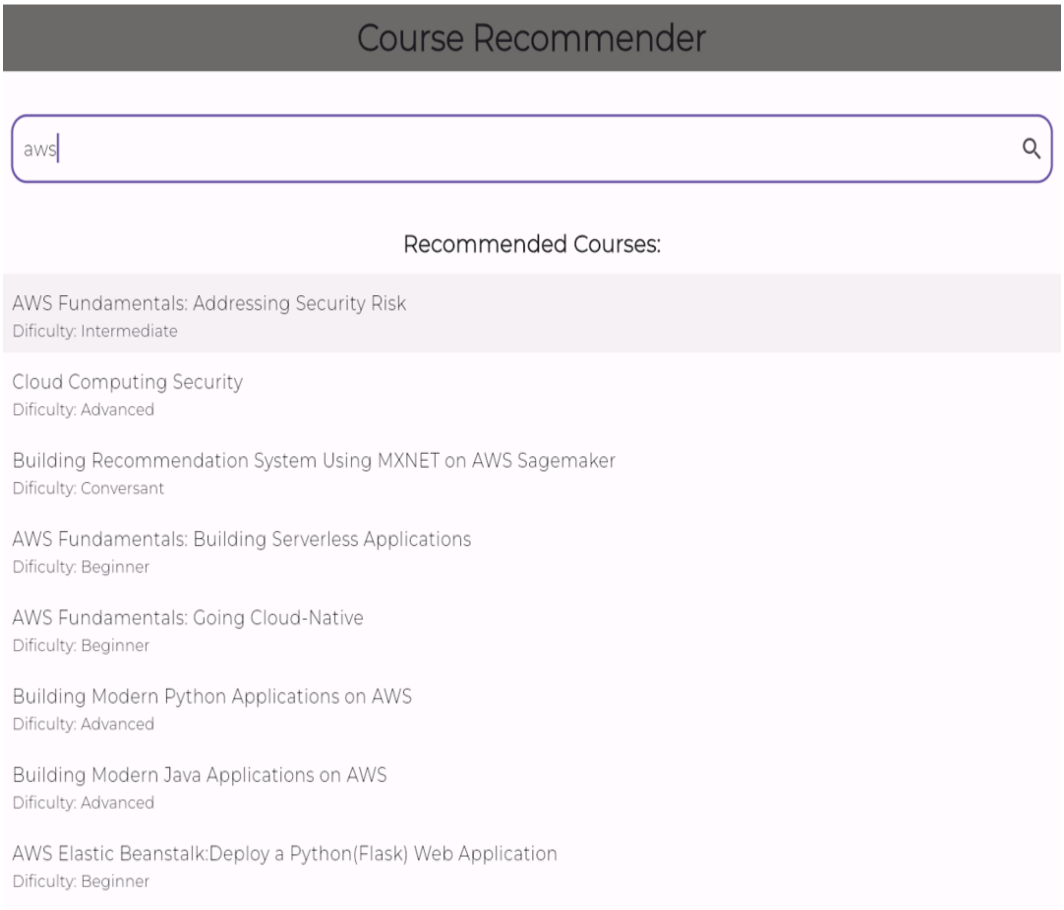


Figure 5.7: **System Test Output**

The input would be to give the user's interested course in the search bar and in the android studio we can see that the search bar being responsive and the server is accepting the requests and the courses are sent and the recommended results are returned in JSON format to the UI.

5.3.4 Test Result

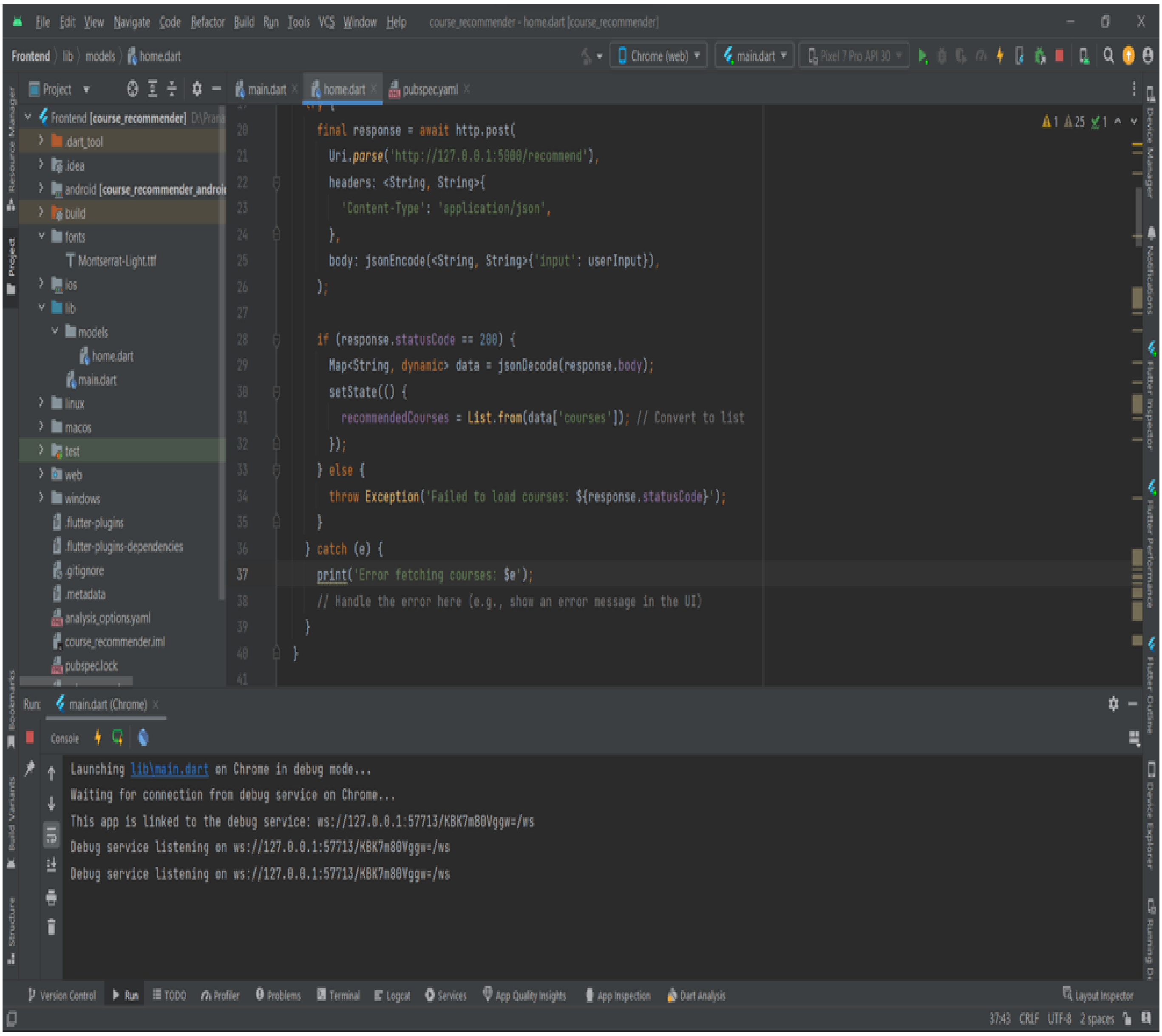


Figure 5.8: **System Testing Result**

Figure 5.8, describes the fully functional course recommendation system that gives a request to the server and the logs can be observed in the log manager and the request taken by the server is also shown.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system uses both KNN algorithm and Vectorization.The course recommender system, employing vectorization (TF-IDF) and the k-Nearest Neighbors (k-NN) algorithm, excels in its simplicity and effectiveness. Through vectorization, it transforms course descriptions into numbers that the computer can understand, capturing essential details in a smart way. The k-NN algorithm then compares these descriptions, finding courses that are most alike based on their content.

In terms of accuracy, this system does a commendable job in suggesting courses that are quite similar to ones you might already enjoy. Let's say you adore courses about coding. This system would pinpoint other coding-related courses that share similar themes or topics. However, its effectiveness might reduce when dealing with newer or less-popular courses, as it relies on existing data. Nonetheless, for widely known courses or popular subjects, it's quite efficient, providing recommendations that align closely with your interests.

6.2 Comparison of Existing and Proposed System

Existing system:(KNN algorithm)

The system using only the k-Nearest Neighbors (k-NN) method is like finding a twin for a course you like by looking at how similar they are. It works well when you have a few courses to compare, quickly finding ones that seem alike. But when the course collection grows, it's like trying to find a matching puzzle piece in a huge pile—it takes longer and might miss some perfect matches.

This method struggles when there are lots of courses because it has to check each one to find the most similar. It's a bit like having to look through a giant bookshelf to find a book that's just right. Sometimes, with all those books, it might not pick the best ones or take a while to give suggestions. So, while it's good for smaller sets of courses, it might get slower and miss out as the collection gets bigger.

Proposed system:(Vectorization and KNN algorithm)

The system using both vectorization (like TF-IDF) and the k-Nearest Neighbors (kNN) algorithm steps ahead as it understands courses like a human reader. Imagine if you're trying to find a book similar to your favorite one. This system doesn't just peek at the book covers it reads and understands the book summaries to recommend ones that feel alike. It's like having a friend who knows exactly the type of books you love!

By turning words into numbers (vectorization), it gets a better grip on what courses are all about. Then, by looking at how courses relate to each other (k-NN), it gives more spot-on suggestions. While it's a bit more complicated at the start, it ends up offering more precise and helpful recommendations, especially when there's a vast library of courses to explore.

6.3 Sample Code

main.py

```
from flask import Flask, request, jsonify
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import NearestNeighbors
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

# Sample course data (replace with your dataset)
course_data = pd.read_csv('Dataset1.csv')
# Create a DataFrame from course data
courses_df = pd.DataFrame(course_data)

# Vectorize the course descriptions
tfidf = TfidfVectorizer(stop_words='english')
course_vectors = tfidf.fit_transform(courses_df['Course Description'])
# print(course_vectors)

# Build the k-Nearest Neighbors model
knn_model = NearestNeighbors(n_neighbors=100, algorithm='brute', metric='cosine')
knn_model.fit(course_vectors)

@app.route('/recommend', methods=['POST'])
```

```python
def recommend_courses():
    user_input = request.json.get('input')
    if request.method == 'POST':
        if user_input:
            # Transform user input into TF-IDF vector
            user_tfidf = tfidf.transform([user_input])

            # Find k-nearest neighbors for user input
            _, indices = knn_model.kneighbors(user_tfidf)

            # Retrieve recommended courses based on nearest neighbors
            recommended_courses = [{'title': courses_df.iloc[idx]['Course Name'],
                                    'description': courses_df.iloc[idx]['Course Description'],
                                    'difficulty': courses_df.iloc[idx]['Difficulty Level'],
                                    'rating': courses_df.iloc[idx]['Course Rating'],
                                    'link': courses_df.iloc[idx]['Course URL']}
                                    for idx in indices[0]]

            return jsonify({'courses': recommended_courses})
        else:
            return jsonify({'error': 'Invalid input'}), 400
    else:
        return jsonify({'error': 'Method Not Allowed'}), 405

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

Output

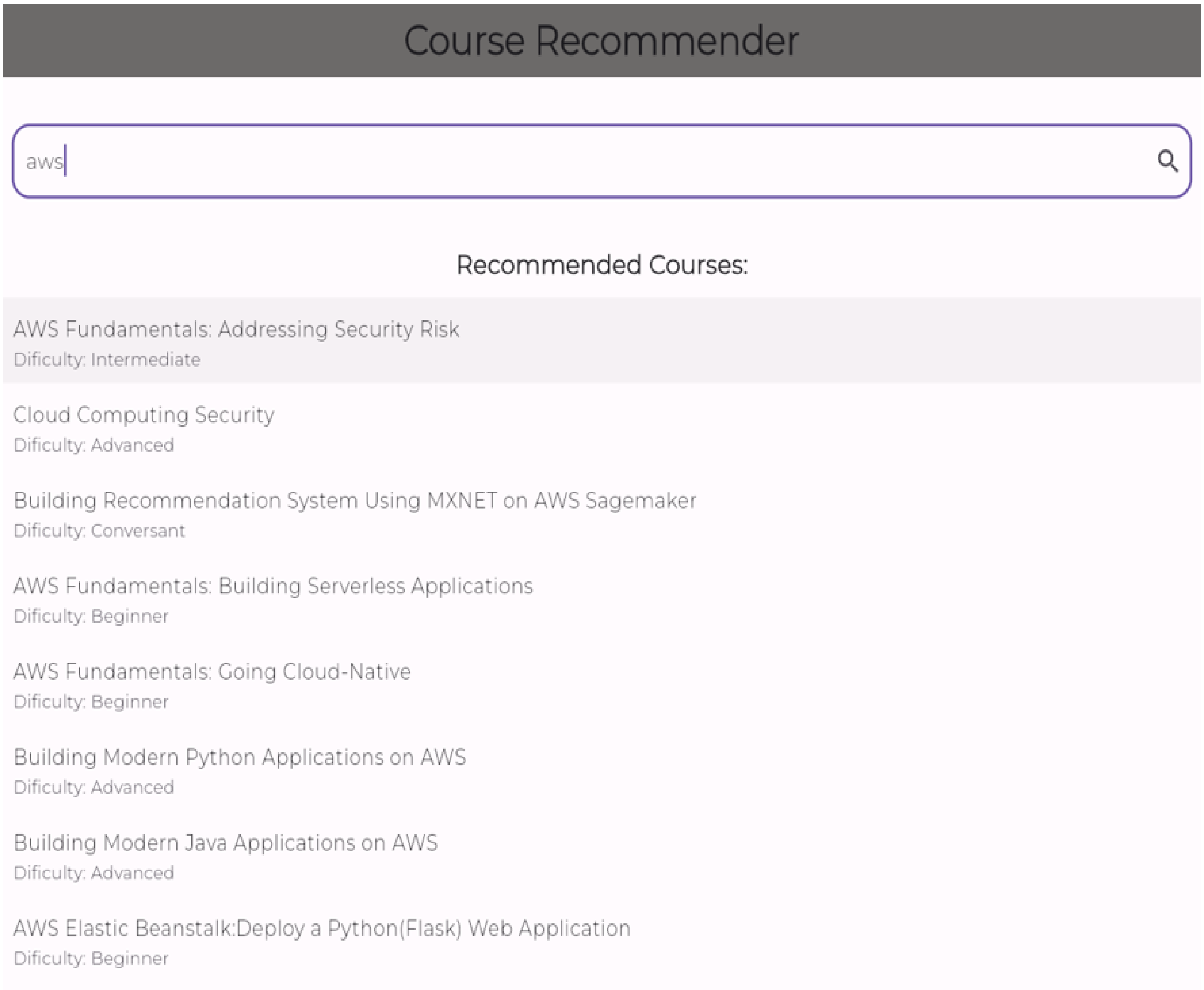


Figure 6.1: **Output**

Figure 6.1 depicts the recommended courses for the topic ”CSS” and the results are obtained in the UI.The ouput of the courses also include the description, ratings and the link to the course.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Our course recommendation project acts like a knowledgeable assistant guiding users to discover courses aligned with their interests. It's akin to a savvy companion who comprehends your preferences and offers suggestions based on that understanding. Using diverse technologies, we crafted a smart backend responsible for understanding your preferences, while the frontend, the part you see and interact with on your device, presents these recommendations in a user-friendly manner.

Behind the scenes, the backend employs complex methods to analyze your input and find relevant courses. The frontend, on the other hand, is designed to be intuitive and easy to use, displaying these tailored suggestions seamlessly. The aim is to simplify the process of discovering courses that resonate with your interests, making the journey of exploring educational opportunities more engaging and personalized, akin to having a knowledgeable guide at your fingertips.

The course recommendation system, employing a combination of TF-IDF vectorization and k-Nearest Neighbors (k-NN) algorithm, provides valuable insights into recommending online courses based on textual similarity. It offers an effective means to assist users in discovering courses aligned with their interests. However, while functional, there are areas for improvement.

7.2 Future Enhancements

In the future, we could make our course recommendation system even better! Imagine adding features that understand not only what courses you like but also why you like them. We could personalize suggestions by considering your past feedback or the specific skills you want to develop. Another cool idea is including more interactive elements, like quizzes or progress tracking, to make learning feel like a fun journey.

Making the app available offline would be awesome too, so you can explore courses even without an internet connection. Lastly, we could expand beyond just courses, recommending books, articles, or videos related to your interests. These enhancements aim to make the system smarter, more engaging, and tailored to your unique learning needs, making discovering new things a delightful experience!.In essence, these enhancements aim to elevate the system by making it more intuitive, versatile, and tailored to individual learning needs, fostering a more engaging and enriching learning journey for users.

1.Enhanced User Interface: Develop an intuitive and user-friendly interface to improve user interaction and engagement.

2.Advanced Algorithms: Explore and integrate advanced recommendation algorithms like matrix factorization or neural collaborative filtering for more accurate and diverse suggestions.

3.Personalization and Feedback Loop: Implement user feedback mechanisms to enhance recommendations based on user preferences over time, enabling a more personalized experience.

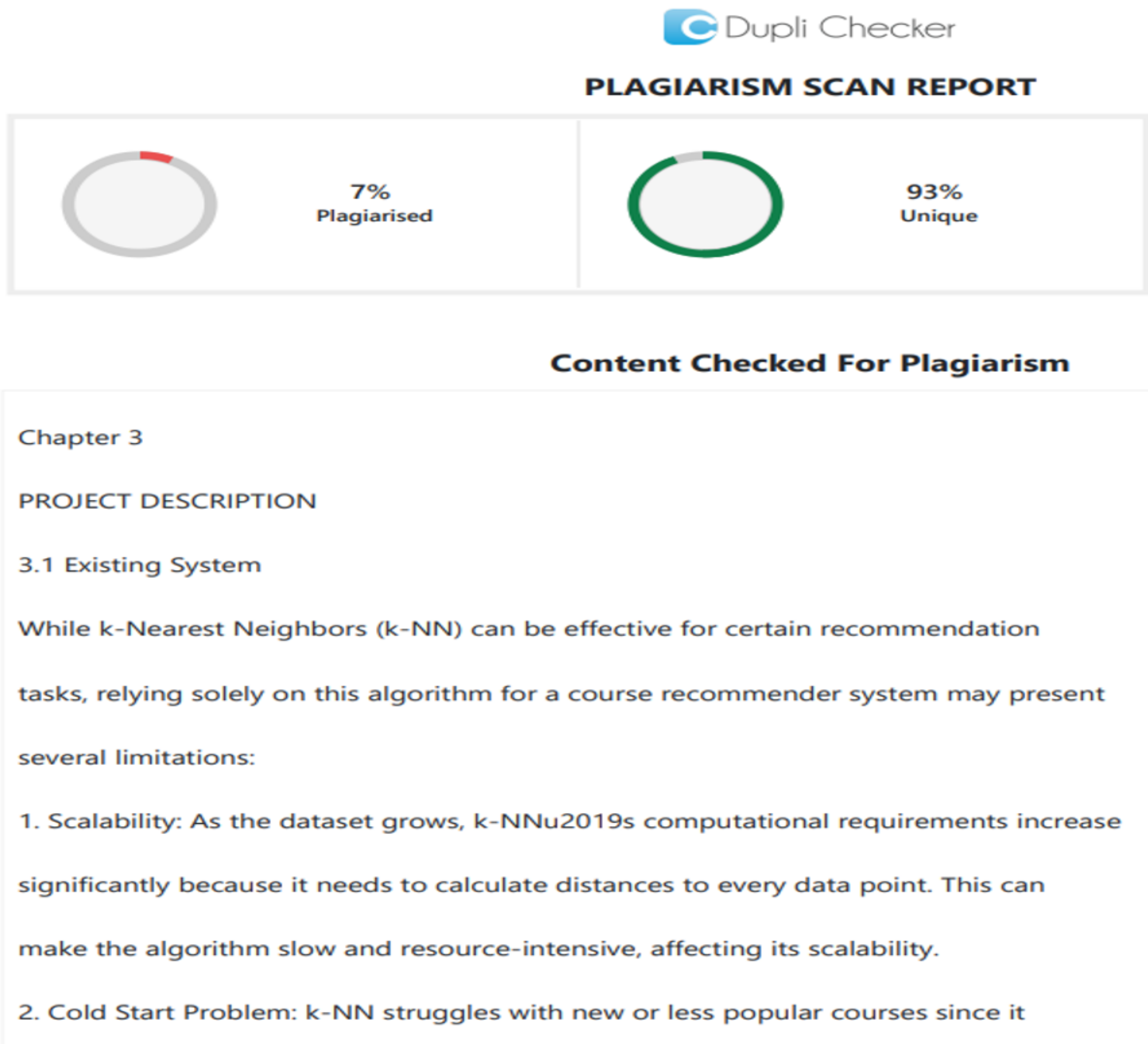
4.Real-time Updates: Incorporate mechanisms to dynamically update the recommendation model with new course data to ensure relevancy.

5.Content Enrichment: Incorporate additional data sources or multimedia content to enrich course descriptions for better analysis.

By focusing on these enhancements, the course recommendation system can evolve into a more robust and tailored solution, offering users a more comprehensive and engaging platform to explore and discover online courses.

Chapter 8

PLAGIARISM REPORT


Dupli Checker

PLAGIARISM SCAN REPORT

7%
Plagiarised

93%
Unique

Content Checked For Plagiarism

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

While k-Nearest Neighbors (k-NN) can be effective for certain recommendation tasks, relying solely on this algorithm for a course recommender system may present several limitations:

1. Scalability: As the dataset grows, k-NNu2019s computational requirements increase significantly because it needs to calculate distances to every data point. This can make the algorithm slow and resource-intensive, affecting its scalability.

2. Cold Start Problem: k-NN struggles with new or less popular courses since it

Figure 8.1: **Plagiarism**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

main.py

```
from flask import Flask, request, jsonify
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import NearestNeighbors
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

# Sample course data (replace with your dataset)
course_data = pd.read_csv('Dataset1.csv')
# Create a DataFrame from course data
courses_df = pd.DataFrame(course_data)

# Vectorize the course descriptions
tfidf = TfidfVectorizer(stop_words='english')
course_vectors = tfidf.fit_transform(courses_df['Course Description'])
# print(course_vectors)

# Build the k-Nearest Neighbors model
knn_model = NearestNeighbors(n_neighbors=100, algorithm='brute', metric='cosine')
knn_model.fit(course_vectors)

@app.route('/recommend', methods=['POST'])
def recommend_courses():
    user_input = request.json.get('input')
    if request.method == 'POST':
        if user_input:
            # Transform user input into TF-IDF vector
            user_tfidf = tfidf.transform([user_input])

            # Find k-nearest neighbors for user input
            _, indices = knn_model.kneighbors(user_tfidf)
```

```python

            # Retrieve recommended courses based on nearest neighbors
            recommended_courses = [{'title': courses_df.iloc[idx]['Course Name'],
                                    'description': courses_df.iloc[idx]['Course Description'],
                                    'difficulty': courses_df.iloc[idx]['Difficulty Level'],
                                    'rating': courses_df.iloc[idx]['Course Rating'],
                                    'link': courses_df.iloc[idx]['Course URL']}
                                    for idx in indices[0]]

            return jsonify({'courses': recommended_courses})
        else:
            return jsonify({'error': 'Invalid input'}), 400
    else:
        return jsonify({'error': 'Method Not Allowed'}), 405

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

9.2 Poster Presentation

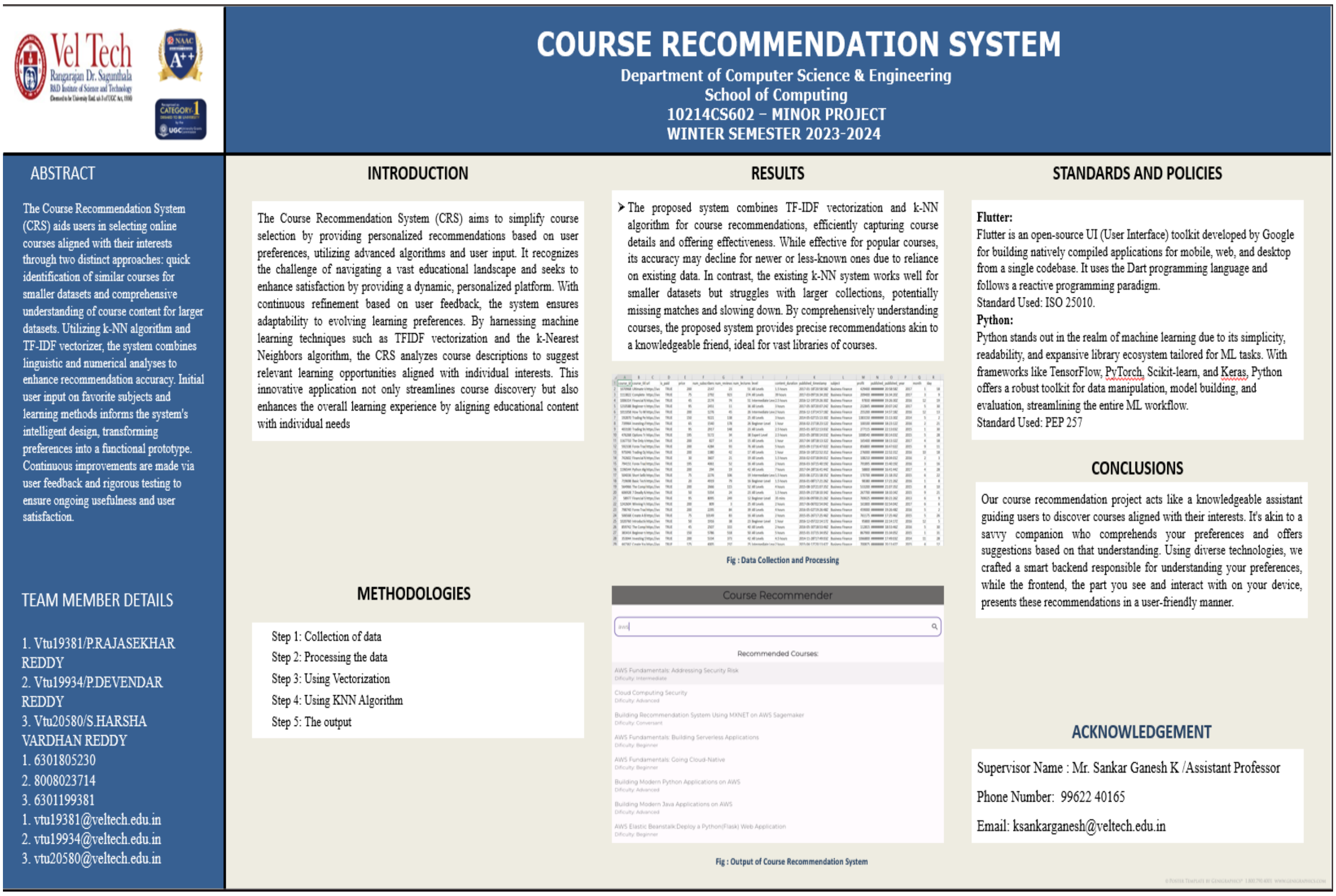


Figure 9.1: **Poster**

References

[1] B. Mondal, O. Patra, S. Mishra and P. Patra, "A course recommendation system based on grades," International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, pp. 1-5, 2020

[2] Derick Prince B, Madhan K, Vishwa K and Yamunathangam D, "Job and Course Recommendation System using Collaborative Filtering and Naive Bayes algorithms," 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, pp. 1-4, 2023.

[3] H. Aoulad Ali, C. Mohamed, B. Abdelhamid and T. El Alami, "A Course Recommendation System for Moocs Based On Online Learning," XI International Conference on Virtual Campus (JICV), Salamanca, Spain, pp. 1-3, 2021.

[4] M. I. Emon et al., "Profile Based Course Recommendation System Using Association Rule Mining and Collaborative Filtering," International Conference on Science Contemporary Technologies (ICSCT), Dhaka, Bangladesh, pp. 1-5, 2021.

[5] M. Isma'il, U. Haruna, G. Aliyu, I. Abdulmumin and S. Adamu, "An Autonomous Courses Recommender System For Undergraduate Using Machine Learning Techniques," International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Ayobo, Nigeria, pp. 1-6, 2020.

[6] M. M. Rahman, M. S. Islam, R. R. Richi and A. Chakraborty, "Course Recommendation System for Students Using K-Means and Association Rule Mining," International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, pp. 641-646, 2022

[7] P. Mishra and V. Jain, "Course Recommendation System using Content-based Filtering," 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 1431-1436, 2023.

[8] S. Lazarevic, T. Zuvela, S. Djordjevic, S. Sladojevic and M. Arsenovic, "Machine learning driven course recommendation system," 21st International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, pp. 1-5, 2022

[9] V. Sankhe, J. Shah, T. Paranjape and R. Shankarmani, ”Skill Based Course Recommendation System,” IEEE International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, pp. 573-576, 2020.

[10] X. Pan, X. Li and M. Lu, ”A MultiView Courses Recommendation System Based on Deep Learning,” International Conference on Big Data and Informatization Education (ICBDIE), Zhangjiajie, China, pp. 502-506, 2020.