" Career Recommendation System "

A Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Technology in *Computer Science & Engineering*

Submitted by:

Aanchal Patel	*0827CS201002*	***Guided by:***
Aditi Rathore	*0827CS201013*	***Prof. Priyanka Jangde***
Aman Solanki	*0827CS201027*	

Acropolis Institute of Technology & Research, Indore

january-june 2023

EXAMINER APPROVAL

The Project entitled ***"Career Recommendation"*** submitted by **Aanchal Patel (0827CS201002), Aditi Rathore (0827CS201013), Aman Solanki (0827CS201027)** has been examined and is hereby approved towards partial fulfillment for the award of **Bachelor of Engineering degree in Computer Science & Engineering** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date:21/04/2023 **Date:**

RECOMMENDATION

This is to certify that the work embodied in this project entitled **"*Career Recommendation*"** submitted by **Aanchal Patel (0827CS201002), Aditi Rathore (0827CS201013), Aman Solanki (0827CS201027)** is a satisfactory account of the bonafide work done under the supervision of **Dr. Kamal Kumar Sethi,** is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science & Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide) **(Project Coordinator)**

STUDENTS UNDERTAKING

This is to certify that a project entitled **"*Career Recommendation*"** has developed by us under the supervision of Prof. Priyanka Jangde. The whole responsibility of work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

Aanchal Patel (0827CS201002)
Aditi Rathore (0827CS201013)
Aman Solanki (0827CS201027)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are a number of people without whom this project's work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentors **Dr. Kamal Kumar Sethi,** Associate Professor, AITR, for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say, “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

Aanchal Patel (0827CS201002), Aditi Rathore (0827CS201013), Aman Solanki (0827CS201027)

Executive Summary

Career Recommendation

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Computer Science & Engineering branch under the sagacious guidance and vigilant supervision of ***Dr. Kamal Kumar Sethi.***

The project is based on deep learning, a branch of machine learning that studies artificial neural networks, which are algorithms inspired by the structure and operation of the brain. TensorFlow, an open-source software library developed by Google for machine learning applications, is used in the project. It is used to identify various characteristics of a person along with there interest's to help recommend a career path. The project makes use of a model that has already been trained on the Microsoft Common Objects in Context (COCO) data set, which roughly contains all common objects. This project's goal is to help individuals make informed decisions about their career paths. Career recommendations can help people identify suitable career options based on their skills, interests, and experience, and provide them with information about job trends, expected salaries, educational requirements, and growth opportunities. Ultimately, the goal is to help individuals find fulfilling, productive, and rewarding careers that align with their strengths, aspirations, and values.

“The only way to

do great work is to love

what you do.

If you haven't found it yet,

keep looking.

Don't settle.”

— Steve Jobs

List of Figures

Table of Contents

Chapter 1 . Introduction

Introduction

The significant growth in the amount of available digital information and the numerous career paths that are available at today's date have created a potential challenge for students to make a firm decision ,as the number of choices increases the confusion to choose the right decision also increases. So, the recommendation system helps the student to explore fields related to education and career opportunities that are available. It narrows down the information based on their interest and makes it easy for them to select their next step.

The Recommender Systems is an application of machine learning in which the model learns from the given data and hence proposes a new recommendation. The recommendation system gives the option to comprehend an individual's taste and find the new, appropriate choice for them consequently. However individuals' preferences shift, they do-follow patterns. Individuals will in-general like things which are comparable to social individual preferences.Our aim is to make a career recommendation system to help students in making a better decision for their future.

1.1 Overview

The student career guidance system can be used by the student to select the best suited career path based on their unique qualities, strengths and weaknesses. This software can help students to select their career path without putting in more effort. Nowadays the number of related study fields is available which makes the student confused to avoid their problems.

We developed this software. The software will recommend only the basis of student response and then follow the same as the previous one . if the student is selected correctly the system will predict the related study fields which would be the final suggestion of the student. a further student wants to change the study field then fill the required details . The number of related study programs is continuously increasing, if further decomposition of study fields into specific sub-fields might be necessary in order to have more specialized fields of study.

1.2 Background and Motivation

Our project is meant to help students make decisions . We use a machine learning program that asks the students questions, and recommends the better stream based on the skills and academic performance provided. Machine learning provides a better ability to upscale, upgrade and obtain results than hard coded algorithms. A machine learning model is an entity that understands the problem – this is obviously better for non-deterministic, real world problems like recommender systems, compared to a pre-programmed system that can do nothing but go by the book. Intuitively, ML is the right approach for this problem, and we have made use of the same.

1.3 Problem Statement and Objectives

The problem is that many individuals struggle with choosing a career path that aligns with their skills, interests, and values, leading to job dissatisfaction, stress, and lack of fulfillment. The vast number of career options and changing job market further contribute to this problem. Therefore, there is a need for an effective career recommendation system that can help individuals identify and explore suitable job opportunities based on their unique preferences and strengths. Such a system should provide personalized guidance, reliable information, and support to aid individuals in making informed decisions about their career paths.

Thus, the system implemented has the following objectives :

1. **Objective 1:** Career choice is a critical moment for any student.Our main objective is to provide the best career option for students on the basis of their skills ,hobbies ,areas of interest and skill.We are making a recommendation system with the help of machine learning models like collaborative filtering

2. **Objective 2:** The recommendation system helps the student to explore fields related to education and career opportunities that are available. It narrows down the information based on their interest and makes it easy for them to select their next step. Individual's educational needs will differ on their career objectives and skills.

1.4 Scope of the Project

This project are made for all the individual who are taking the next step for their futures.This project is gonna help:

- Higher secondary students who need to make the best possible career choice for their future and proceed in that direction.
- Individuals who are trying to start a new career as they may be tired of their current situation .
- People who are starting their career very late because they may not able to build their career at early age maybe cause of poor financial status or

our society norms for example generally women who are still repressed at some corner of our country

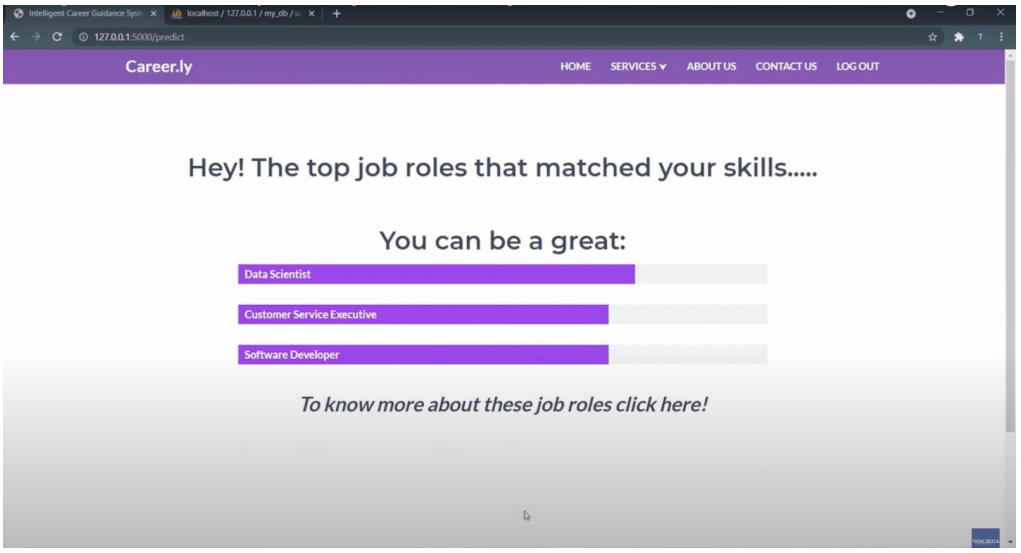

Figure 1-1: Probability of your interest

1.5 Team Organization

Aanchal Patel:

I investigated and found the right technology and studied it. For the implementation of the project , I collected the object data and trained the model for it. Implementation logic for the project objective and coding of internal functionalities is also done by me. Also, worked on Back end design for storing results in the database for maintaining logs and machine algorithms for the dataset.

Aditi Rathore:

I collected all the necessary information about our project and the working of counting of objects in the project. I also did the front-end as the framework and design ,surveyed various research papers and posters. Documentation is also a part of the work done by me in this project.

Aman Solanki:

Along with doing preliminary investigation and understanding of the current system, I studied the topic and its scope. research papers related to the scholarship and the technology that is to be used. I also worked to find various scholarships data and the working of counting of objects in the project.

1.6 Report Structure

The project **Career Recommendation** is primarily concerned with the **advice or guidance to individuals about career paths or options based on their skills, interests, abilities, and experience** and the whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with a report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of the existing system and highlights the issues and challenges of the project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrates the software engineering paradigm used along with different design representations. The chapter also includes a block diagram and details of major modules of the project. Chapter also gives insights of different types of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interfaces designed in the project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of the project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2 .Review of Literature

Review of Literature

Recommender systems are the systems that are designed to recommend things to the user based on many different factors. These systems predict the most likely product that the users are most likely to purchase and are of interest to. Companies like Netflix, Amazon, etc. use recommender systems to help their users to identify the correct product or movies for them.The aim of these recommender system to help user to choose or make decision . A career recommendation system is basically a system that will help students to choose their career path or field based on their preferences and interest.

There are several techniques to make these kinds of recommendation systems.One of the techniques is collaborative filtering. It is also known as the user-to-user correlation method, finds similar users who have the same taste as the target user and recommends items based on what the similar users like. The key step in collaborative filtering is computing the similarities among users. Collaborative filtering recommendation algorithms can be classified into memory-based and model-based.

2.1 Preliminary Investigation

2.1.1 Current System

Traditional career counseling is done at a career center wherein the person has to give many aptitude tests, which are then scored manually. These scores are then reviewed by a career counselor who then assigns you a career stream. This whole process is long, tiresome and error prone. Our system automates this process, resulting in a better error-free career recommendation system, since our recommendation algorithm also takes into account the user's personality, which is a very important factor when deciding someone's career. Aptitude & Academic/Technical skills alone are

not the only deciding factor when it comes to a career decision, Personality and Aptitude both are important.

2.2 Limitations of Current System

- The limitations of these are as follows :
- This whole process is long, tiresome and error prone.
- The model can only make recommendations based on the existing interest of a user.
- In other words, the model has limited ability to expand on the user's existing interests.
- In an item based model, Side Feature Doesn't have much importance.
- Here Side features can be student names or parents name in the context of career recommendation system.

2.3 Requirement Identification and Analysis for Project

Significant work has been done in the field of Career Recommendation ; however, it is not easy to achieve desired results. The review of literature leads to draw certain major findings which are as under :

Select the student area of interest and get all the subjects with marks.
Administrators already assigned courses for each subject.
Students choose physics, math, chemistry subjects and compare them with threshold values. If student got above threshold value means suggest that engineer as career Otherwise the system goes for the next category and so on.[1]

After the analysis appropriate recommendation of a particular domain or career choice listing out all the possibilities of success rates in percentile for that particular field of domain and comparing it with the profile of the user concerning a similar database.[2]

The paper focused on the problem of learning to recommend interests from a small training database. The performance depends crucially on the features that are used to represent the test. Specifically, it shows that using local edge orientation histograms (EOH) as features can significantly improve performance compared to the standard linear features used in existing systems. For probability, local orientation histograms enable state of the art performance using only a few hundred training examples.[3]

`

The Data Mining techniques applied to data to enhance the features. Building a recommendation system considering cold start problem. Building a hybrid recommendation system considering different algorithms and their accuracy. [4]

Several recommendation systems have been anticipated are based on collaborative filtering and so far most of them have been able to resolve the problems while providing improved recommendations. Most of them have been able to resolve the problems while providing improved recommendations.[5]

Different types of recommender systems that could be implemented and the different types of algorithms that could be used for better accuracy.[6]

Recommender System (RS) can be defined as programs and techniques that provide suggestions for items such as products or services that are most likely of interest to a particular user namely individuals or businesses. Recommender Systems (RS) can be defined as programs also techniques that provide suggestions for items such as products or services that are most likely of interest to a particular user namely individuals or businesses.[7]

Reducing overload information by retrieving the most relevant
information and services from a big amount of data is the aim of developing recommender systems, therefore, it is hoped can provide personalized services.[8]

The most important thing about a recommender system is its capability to predict a user's preferences and interests by analyzing the behavior of this user and/or the behavior of other users to generate personalized recommendations[9]

This paper proposes a professional suggestion framework utilizing brain networks because of the significant number of boundaries for characterization. These boundaries remember understudy execution for different subjects present in the undergrad educational program of software engineering and understudy interests, relational abilities, gifts, and so on[10].

2.3.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. This project presents Student Career Guidance and Recommendation System using the inherent student skills for choosing the right career. Choosing a right career is significant due to the diversified human abilities. Many students are choosing their career path without receiving proper advice from suitable professional or university services. This may potentially cause mismatch between academic achievements, personality, interest and abilities of the students. In order to recommend students in career selection, it is essential to build a recommendation system that provides direction and guidance to students in choosing their career. The key challenge in this project is selecting key attributes/skills that help in predicting the right path to meet diversified students goals. Here, we have developed a software tool to evaluate the aptitude and personality of a person based on his/her academic level using carefully curated personality and aptitude tests. This tool will help you determine your aptitude and personality traits, and will eventually help you in choosing your own career path.

Chapter 3. Proposed System

Proposed System

3.1 The Proposal

The proposed model is a machine learning model that generates recommendations' using recommender systems concept of machine learning by analyzing the huge amounts of data assembled using big data technologies. The model would be divided into three major phases of development. The three models clearly define the process of data collection, analysis, classification & recommendation. The model analyzes the metrics in a very curated way and could suggest the user or an organization with a recommendation that could be early advice and could help in making better decisions at a given point of time. Also, it would result in making the required changes to achieve the goal even before starting the process so that there are no errors in the early phases of the process.

3.2 Benefits of the Proposed System

A career recommendation system can provide several benefits to individuals who are seeking guidance in choosing a career path. Some of the benefits of a career recommendation system include:

1. Personalized recommendations: A career recommendation system takes into account an individual's skills, interests, personality traits, and other relevant factors to provide personalized career recommendations. This can help individuals discover career options that align with their strengths and preferences, increasing the likelihood of job satisfaction and success.

2. Time and effort savings: Exploring different career options can be time-consuming and overwhelming. A career recommendation system can streamline the process by providing curated suggestions based on an individual's profile, saving time and effort in researching and evaluating numerous career options.

3. Increased career satisfaction: When individuals are matched with careers that align with their interests, skills, and values, they are more likely to experience higher job satisfaction. A career recommendation system can help individuals make informed decisions about their career path, leading to greater career satisfaction in the long run.

4. Enhanced self-awareness: Career recommendation systems often include assessments or questionnaires that help individuals reflect on their strengths, interests, and personality traits. This can lead to increased self-awareness and a better understanding of one's own abilities and career preferences, which can be valuable in making informed career choices.

5. Expanded career options: A career recommendation system may suggest career options that individuals may not have considered or been aware of, based on their unique profile. This can broaden their horizons and open up new possibilities, helping individuals explore a wider range of career options and make more informed decisions.

6. Improved decision-making: Career decisions are often complex and require careful consideration of multiple factors. A career recommendation system can provide individuals with relevant information and insights, helping them make more informed and data-driven decisions about their career path.

7. Long-term career planning: A career recommendation system can provide guidance not just for immediate career choices, but also for long-term career planning. By helping individuals identify their strengths and interests early on, a career recommendation system can assist in developing a strategic career plan and setting achievable goals for career growth.

3.3 Block Diagram

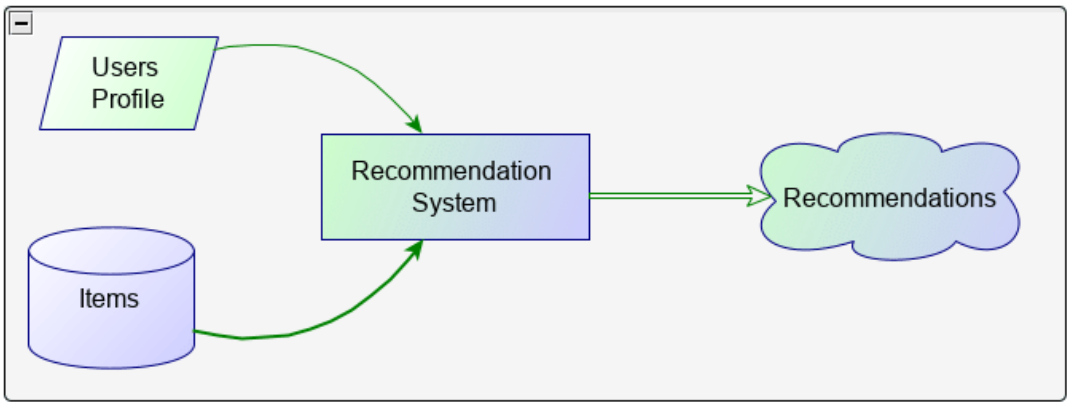

Figure 3-1 : Block Diagram

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

From a technical perspective, building a career recommendation system can be feasible depending on various factors such as data availability, algorithm complexity, and system requirements. Here are some considerations:

1. Data availability: The availability and quality of data can significantly impact the feasibility of a career recommendation system. To provide accurate and relevant recommendations, the system needs access to comprehensive and up-to-date data on careers, job descriptions, skills, interests, and other relevant information. This data can come from various sources such as career databases, labor market data, job postings, and user-generated data. If the data is limited, incomplete, or outdated, it may affect the accuracy and reliability of the recommendations.

2. Algorithm complexity: The complexity of the recommendation algorithm used in the system can also affect its feasibility. More sophisticated algorithms,

such as machine learning algorithms, may require substantial computational resources, expertise in data science, and access to large datasets for training. Developing and implementing complex algorithms may require a higher level of technical expertise and resources, which can impact the feasibility of the system.

3. System scalability and performance: Career recommendation systems may need to handle a large volume of data and users, and should be able to provide recommendations in a timely manner. Ensuring that the system is scalable, efficient, and can handle concurrent requests is essential for its feasibility. It may require robust system architecture, appropriate database management, and optimization techniques to achieve good performance.

4. User interface and experience: The usability and user experience of the system are important factors in its feasibility. The system should have an intuitive and user-friendly interface that is easy to navigate and provides clear recommendations. It should also consider factors such as user privacy, data security, and compliance with relevant regulations, which may impact the technical design and implementation of the system.

5. Maintenance and updates: A career recommendation system may require regular updates to keep the data and algorithms relevant and accurate. Ensuring that the system can be easily maintained and updated without significant disruptions to its functionality is crucial for its feasibility in the long term.

6. Integration with existing systems: Depending on the context, a career recommendation system may need to be integrated with existing systems, such as job portals, career development platforms, or HR management systems. Ensuring seamless integration with existing systems can impact the feasibility of the recommendation system, as it may require technical compatibility and interoperability.

Overall, the feasibility of a career recommendation system from a technical perspective depends on various factors, including data availability, algorithm complexity, system scalability and performance, user interface and experience, maintenance requirements, and integration with existing systems. Proper planning, resource allocation, and technical expertise are necessary to ensure the successful development, implementation, and maintenance of a career recommendation system.

3.4.2 Economical

The feasibility of a career recommendation system from an economic perspective depends on several factors that impact the costs and benefits associated with its development, implementation, and maintenance. Here are some considerations:

1. Development costs: The initial costs associated with developing a career recommendation system can vary widely depending on factors such as the complexity of the system, the technology used, and the expertise required. Costs may include software development, data acquisition and preparation, algorithm development, user interface design, and testing. These costs need to be carefully considered and budgeted for to ensure the economic feasibility of the system.

2. Operational costs: Once the career recommendation system is developed and implemented, there may be ongoing operational costs associated with its maintenance, hosting, data updates, and technical support. These costs should be considered in the economic feasibility assessment, as they can impact the total cost of ownership of the system over its lifespan.

3. Benefits to users: The economic feasibility of a career recommendation system can be evaluated based on the benefits it provides to users. If the system helps individuals make informed career choices, find suitable job opportunities, and enhance their career prospects, it can lead to increased job satisfaction, productivity, and earnings potential. These benefits can contribute to the economic feasibility of the system by creating value for users.

4. Cost savings: A career recommendation system can potentially result in cost savings for users and organizations. For example, by streamlining the career exploration process and providing personalized recommendations, users may save time and effort that would have been spent on research and trial-and-error. Organizations may also benefit from reduced employee turnover, increased job satisfaction, and improved talent matching, which can result in cost savings related to recruitment, training, and retention.

5. Market demand and competition: The economic feasibility of a career recommendation system may also depend on market demand and competition. If there is a significant demand for such systems, and there are few competitors or unique differentiators, it may create a favorable economic environment for the system's success. On the other hand, a saturated market with intense competition may impact the economic viability of the system.

3.4.3 Operational

The operational feasibility of a career recommendation system refers to its ability to be effectively implemented and maintained in an operational environment. Here are some factors to consider when evaluating the operational feasibility of such a system:

1. System scalability: A career recommendation system needs to be able to handle the scale of users and data it is designed to accommodate. As the system grows and more users join, it should be able to handle the increased load and perform optimally without significant degradation in performance. The system should be designed to scale horizontally, allowing for additional resources to be added as needed to meet increasing demand.

2. Data availability and quality: The operational feasibility of a career recommendation system relies heavily on the availability and quality of data. The system requires access to relevant and up-to-date data, such as job market trends, industry information, job descriptions, and user profiles. Ensuring the data is accurate, reliable, and comprehensive is critical to the system's effectiveness and operational feasibility.

3. Integration with existing systems: If the career recommendation system needs to integrate with other existing systems, such as human resource management systems, job boards, or other career-related platforms, its operational feasibility depends on the ease and effectiveness of such integrations. Compatibility and interoperability with existing systems should be assessed to ensure smooth operations and data flow between systems.

4. User interface and user experience: The user interface (UI) and user experience (UX) of the career recommendation system are crucial for its operational feasibility. The system should have an intuitive and user-friendly interface that allows users to easily interact with the system, input their information, and receive recommendations in a clear and understandable manner. An efficient and effective UI/UX design can impact the system's usability, adoption, and overall operational success.

5. Technical support and maintenance: The operational feasibility of a career recommendation system also depends on the availability of technical support and maintenance resources. Regular updates, bug fixes, and system enhancements may be required to keep the system operational and effective. Adequate technical support and maintenance resources should be allocated to ensure the system's ongoing performance and reliability.

6. Security and privacy: Ensuring the security and privacy of user data is critical to the operational feasibility of a career recommendation system. The system should be designed with robust security measures to protect user information from unauthorized access, data breaches, and other security threats. Compliance with relevant privacy regulations, such as GDPR or CCPA, should also be ensured to maintain user trust and confidence in the system.

7. Training and user adoption: User adoption and training are important factors in the operational feasibility of a career recommendation system. Sufficient training and support should be provided to users to familiarize them with the system's features, functionalities, and benefits. Encouraging user adoption and regular usage of the system is crucial for its operational success.

3.5 Design Representation

3.5.1 Data Flow Diagram

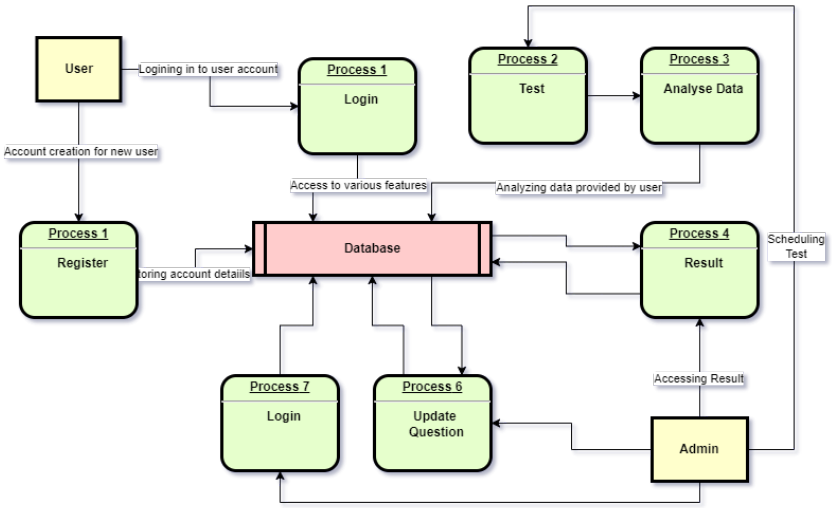

Figure 3-2 : Data Flow Diagram

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

3.6.1 Hardware

1. 32-bit, x86 Processing system
2. Windows 7 or later operating system
3. High processing computer system without GPU or with GPU(high performance).

3.6.2 Software

- HTML 5:Hypertext Markup Language (HTML) is the standard markup language and web applications. IT is used to make the website.
- CSS: IT is used to make the website more attractive.
- Javascript: it is used to make the pages more interactive.
- PHP 5.2.0:PHP is aside scripting language development, and also used as a general purpose programming language.
- Python : Python is used to implement machine learning algorithms.
- MySQL: It is an open source relational database management system .It is used to store and manage databases.

Chapter 4 . Implementation

Implementation

A website to guide an interested career, based on his/her interested area. The Proposed website consists of a set of questions with a list of answers as a MCQ's where these questions extract his skills or interested domain of studies. The proposed model is a machine learning model that generates recommendations' using recommender systems concept of machine learning by analyzing the huge amounts of data assembled using big data technologies.

4.1 Technique Used

4.1.1 KNN Algorithm

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN 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.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

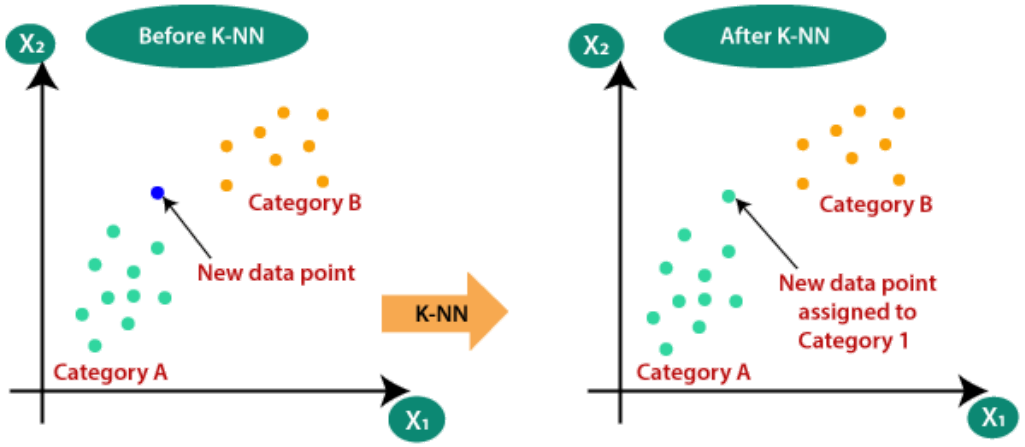

Figure 4-1 : Deep Learning

4.2 Tools Used

4.2.1 HTML 5:

Hypertext Markup Language (HTML) is the standard markup language and web applications. IT is used to make the website. HTML5 is a markup language used for creating and structuring content on the web. It is the fifth version of the Hypertext Markup Language (HTML) and is designed to make it easier to develop rich multimedia content and dynamic web pages. HTML5 includes new elements and attributes that allow developers to create more complex and interactive web applications, such as video and audio, graphics, animations, and advanced forms. HTML5 is supported by all modern web browsers and is being adopted rapidly due to its improved usability, accessibility, and compatibility across devices.

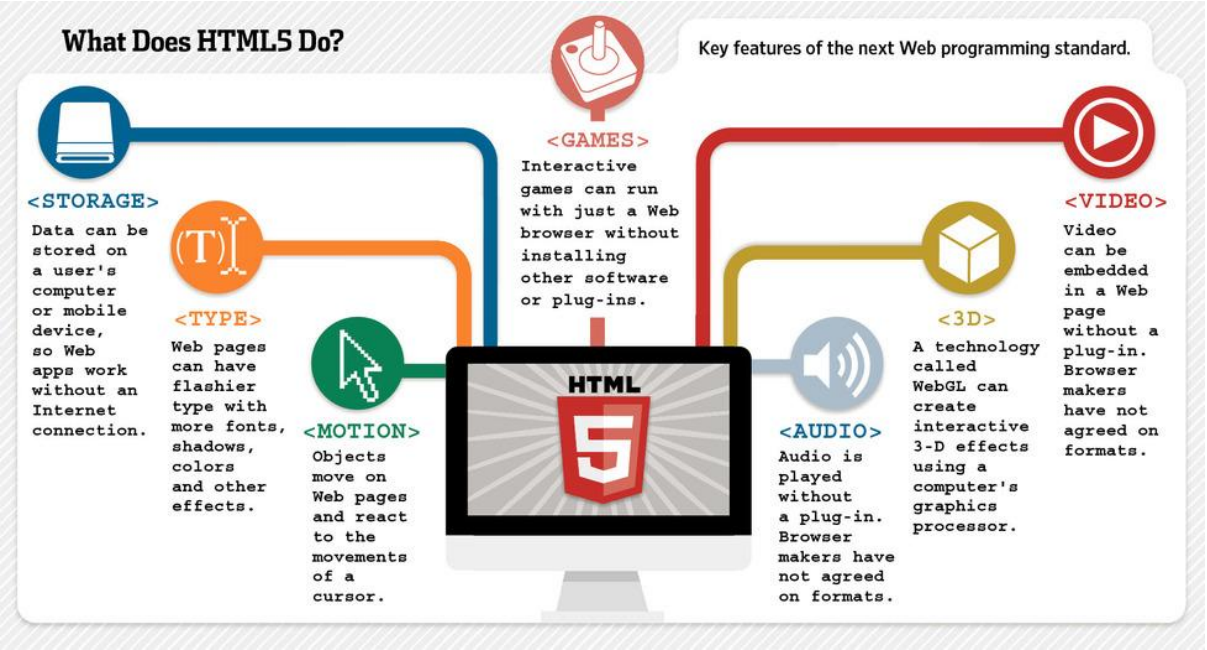

Figure 4-2 : HTML 5 Features

4.2.2 CSS:

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

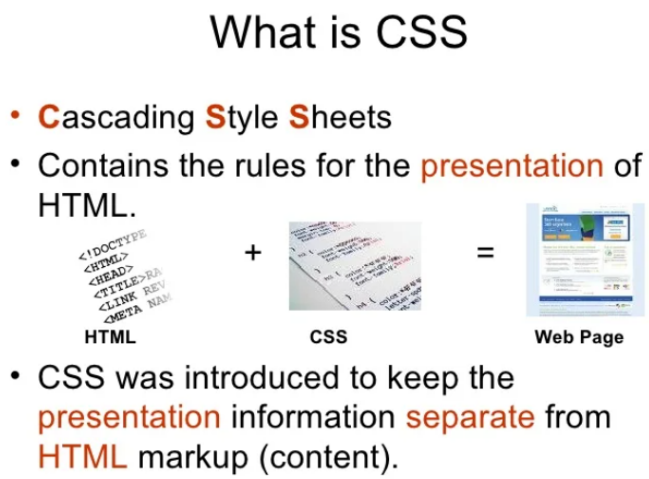

Figure 4-3 : CSS

4.2.3 PHP 5.2.0:

- New memory manager for the Zend Engine with improved performance and a more accurate memory usage tracking.
- Input filtering extension was added and enabled by default.
- JSON extension was added and enabled by default.
- ZIP extension for creating and editing zip files was introduced.
- Hooks for tracking file upload progress were introduced.
- Introduced E_RECOVERABLE_ERROR error mode.
- Introduced DateTime and DateTimeZone objects with methods to manipulate date/time information.
- Upgraded bundled SQLite, PCRE libraries.
- Upgraded OpenSSL, MySQL and PostgreSQL client libraries for Windows installations.
- Many performance improvements.
- Over 200 bug fixes.

Figure 4-4 : PHP 5.2.0

4.2.4 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

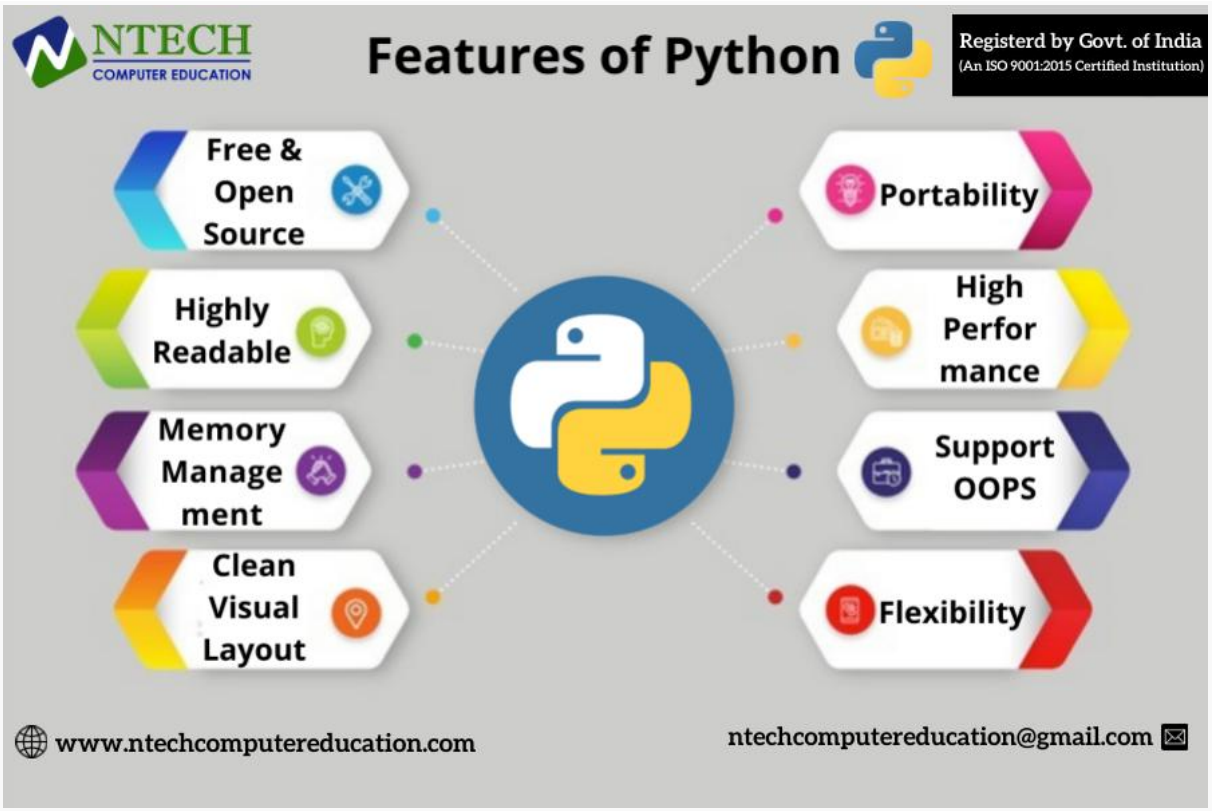

Figure 4-5: Features of Python

4.3 Language Used

Python language is used in the system due to the following Characteristics :

Simple :

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudocode nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source :

Python is an example of a FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

Object Oriented :

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

Extensive Libraries :

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML,

XML-RPC, HTML, WAV files, cryptography, GUI(graphical user interfaces) using Tk, and also other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the "batteries included" philosophy of Python.

4.4 Screenshots

The Following are the screenshots of the result of the project :

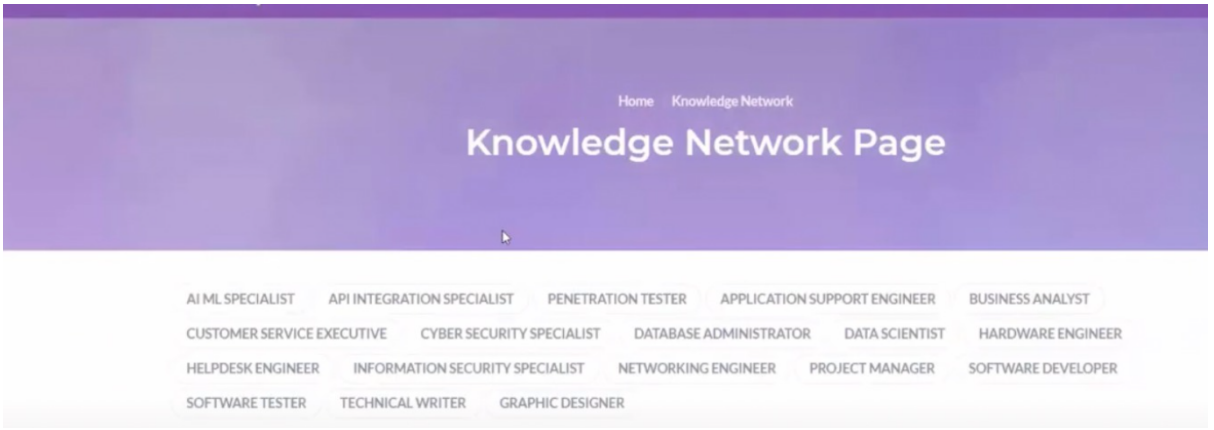

Figure 4-6 : Screenshot 1

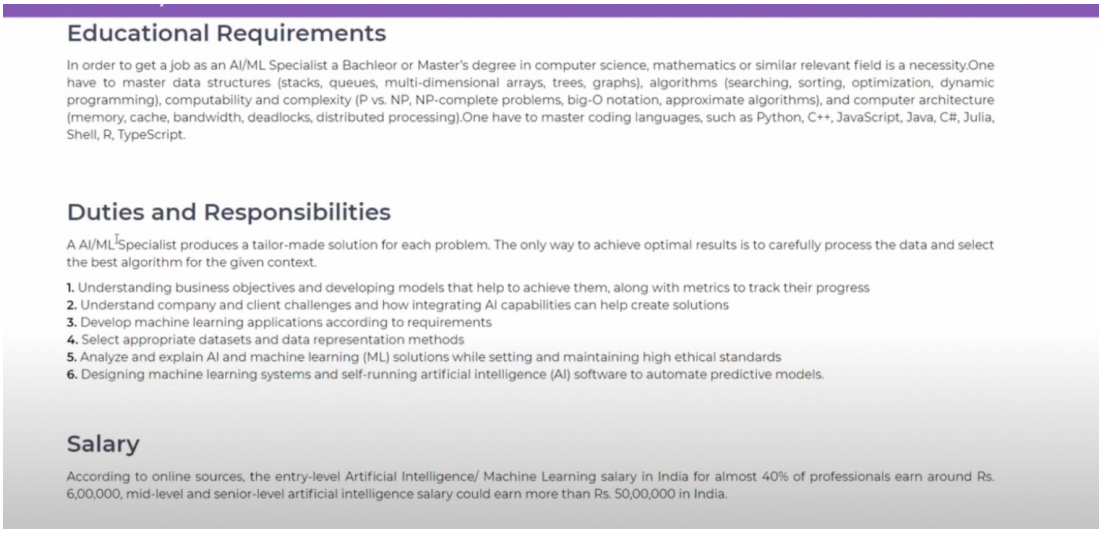

Figure 4-7 : Screenshot 2

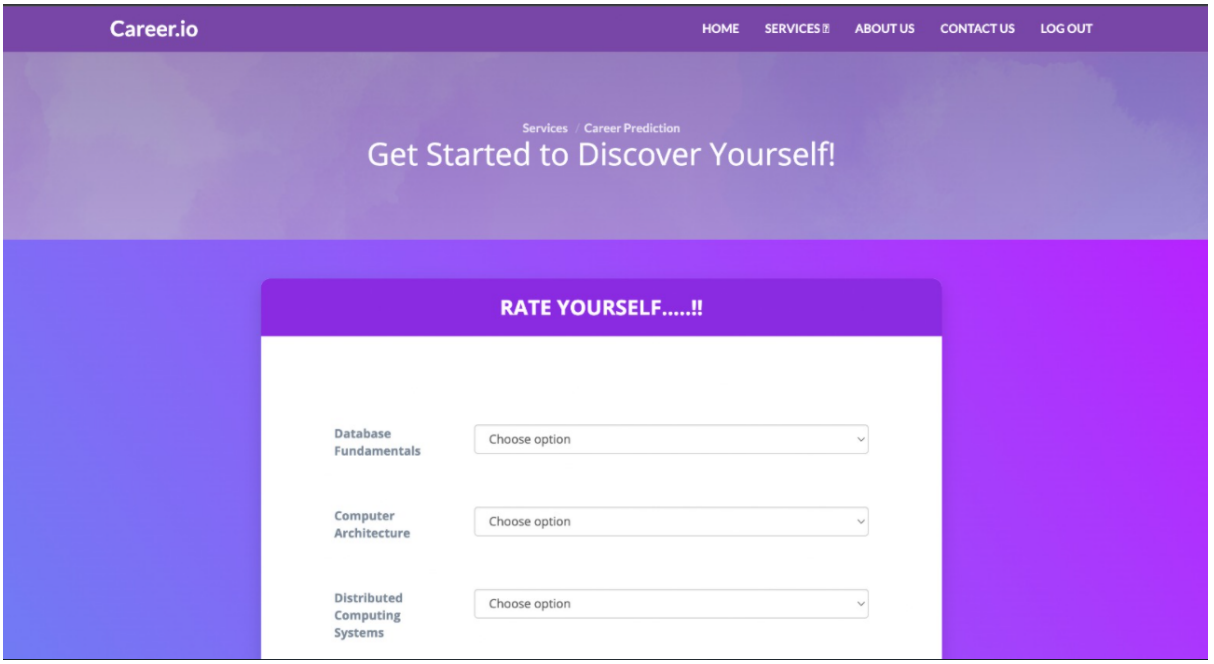

Figure 4-8 : Screenshot 3

4.5 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

4.5.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

4.5.2 Test Case and Analysis

TEST CASE: 1

Test Case ID	TC001
Test Case	It will provide recommendation to the user
Summary	with accuracy >=50% or not.
Test	login and start the test.
Expected	The user will get the probability of their interest with accuracy greater than 50%.
Actual Result	The User with accuracy greater than 50% are detected.
Status	Pass

Table 1 - Case Table-1

TEST CASE 1 OUTPUT

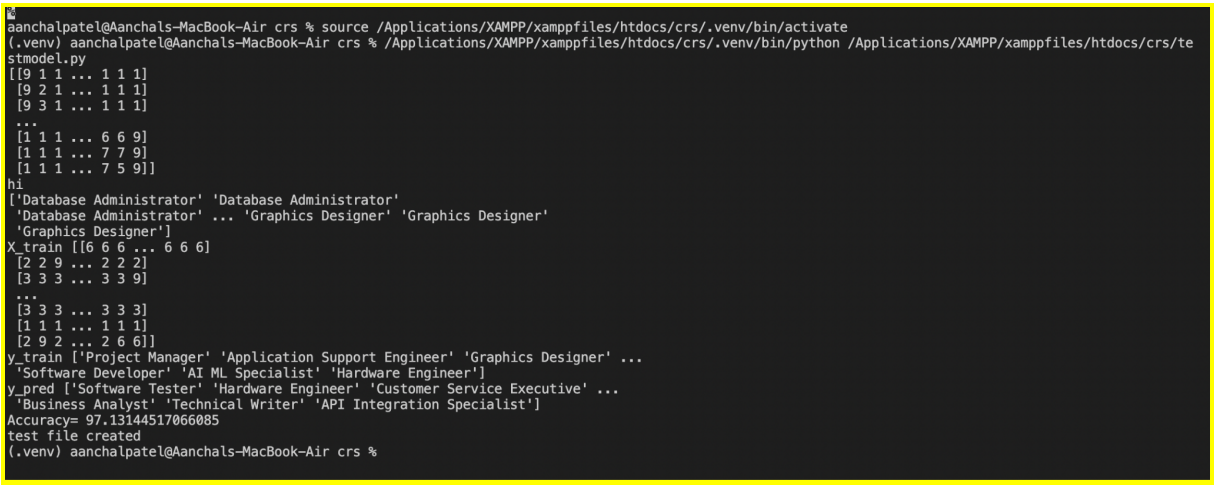

Figure 4-9 : Screenshot 4

Chapter 5.Conclusion

Conclusion

5.1 Conclusion

In conclusion, a career recommendation system can provide significant benefits by helping individuals make informed career choices, matching their skills, interests, and goals with relevant job opportunities. It can streamline the job search process, enhance career planning, and improve job satisfaction. However, like any other technology, career recommendation systems also have limitations and scope for improvement.

In conclusion, while career recommendation systems offer significant benefits, they also have limitations and scope for improvement. Continued advancements in data quality, bias mitigation, contextualization, real-time updates, and user-centric design can contribute to the ongoing improvement and operational feasibility of career recommendation systems, making them more effective in helping individuals make informed career choices.

5.2 Limitations of the Work

Some limitations of career recommendation systems include:

1. Data limitations: The accuracy and reliability of recommendations depend on the availability and quality of data. Limited or inaccurate data may result in less accurate or relevant recommendations.

2. Bias: Career recommendation systems may unintentionally introduce bias, such as gender or race bias, based on the data used for recommendations. This can result in unfair or discriminatory recommendations.

3. Lack of context: Career recommendation systems may not fully capture the complex and nuanced factors that influence career choices, such as personal preferences, cultural factors, or individual circumstances, which may affect the relevance and accuracy of recommendations.

4. Dynamic job market: The job market is constantly evolving, and career recommendation systems may struggle to keep up with the changing trends and dynamics, resulting in recommendations that may not be up-to-date.

5. User preferences and feedback: Career recommendation systems may not always fully capture individual user preferences and feedback, leading to suboptimal recommendations that do not align with the user's unique needs and preferences.

5.3 Suggestion and Recommendations for Future Work

There are several scopes for improvement in career recommendation systems, including:

1. Enhanced data collection and accuracy: Improving data collection methods and data quality can lead to more accurate and reliable recommendations. Incorporating more diverse and comprehensive data sources can also help reduce biases and improve the relevance of recommendations.
2. Addressing bias and fairness: Implementing techniques to detect and mitigate bias in recommendations, such as fairness-aware algorithms and ethical AI practices, can help ensure that recommendations are fair, unbiased, and inclusive.
3. Incorporating contextual information: Incorporating more contextual information, such as personal preferences, cultural factors, and individual circumstances, can result in more relevant and tailored recommendations that better align with users' unique needs.
4. Real-time updates and adaptability: Incorporating real-time updates and adaptability to changing job market trends can help ensure that recommendations remain up-to-date and relevant in the dynamic job market landscape.
5. User-centric design: Continuously gathering user feedback and incorporating user-centric design principles can help improve the user interface, user experience, and overall usability of the career recommendation system.

Bibliography

[1] https://www.ijedr.org/papers/IJEDR1903111.pdf

[2] https://ijarcce.com/wp-content/uploads/2021/09/IJARCCE.2021.10809.pdf

[3] https://www.researchgate.net/publication/341585130_Recommender_System_For_Educational_Analysis_In_Prediction_of_Appropriate_Career_Domain_Recommendations_using_Machine_Learning_Techniques

[4] https://www.technoarete.org/common_abstract/pdf/IJERCSE/v8/i8/Ext_17459.pdf

[5] https://www.researchgate.net/publication/360879407_Trends_and_Characteristics_of_Career_Recommendation_Systems_for_Fresh_Graduated_Students

[6] https://esource.dbs.ie/bitstream/handle/10788/4254/msc_jeevankrishna_2020.pdf?sequence=1&isAllowed=y

[7] https://www.hindawi.com/journals/sp/2022/3437139/

[8] https://ieeexplore.ieee.org/document/9137992

[9] http://nebula.wsimg.com/66d211a546a597e6a5c2a8cfe4424095?AccessKeyId=DFB1BA3CED7E7997D5B1&disposition=0&alloworigin=1

[10] https://ncert.nic.in/tamanna/

[11] https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a

[12] https://www.ijraset.com/research-paper/career-recommendation-using-ann

[13] https://www.ijresm.com/Vol_1_2018/Vol1_Iss10_October18/IJRESM_V1_I10_58.pdf

[14] https://www.ijrte.org/wp-content/uploads/papers/v7i6s4/F11020476S419.pdf

Source Code

Testapp.py

```
from flask import Flask, render_template, request
import pickle
import numpy as np

app = Flask(__name__)


@app.route('/')
def career():
   return render_template("hometest.html")


@app.route('/predict',methods = ['POST', 'GET'])
def result():
  if request.method == 'POST':
     result = request.form
     i = 0
     print(result)
     res = result.to_dict(flat=True)
     print("res:",res)
     arr1 = res.values()
     arr = ([value for value in arr1])

     data = np.array(arr)

     data = data.reshape(1,-1)
     print(data)
     loaded_model = pickle.load(open("careerlast.pkl", 'rb'))
     predictions = loaded_model.predict(data)


     print(predictions)
     pred = loaded_model.predict_proba(data)
     print(pred)
     #acc=accuracy_score(pred,)
     pred = pred > 0.05
     #print(predictions)
     i = 0
     j = 0
     index = 0
     res = {}
     final_res = {}
     while j < 17:
         if pred[i, j]:
             res[index] = j
```

```
            index += 1
        j += 1
    # print(j)
    #print(res)
    index = 0
    for key, values in res.items():
        if values != predictions[0]:
            final_res[index] = values
            print('final_res[index]:',final_res[index])
            index += 1
    #print(final_res)
    jobs_dict = {0:'AI ML Specialist',
                 1:'API Integration Specialist',
                 2:'Application Support Engineer',
                 3:'Business Analyst',
                 4:'Customer Service Executive',
                 5:'Cyber Security Specialist',
                 6:'Data Scientist',
                 7:'Database Administrator',
                 8:'Graphics Designer',
                 9:'Hardware Engineer',
                 10:'Helpdesk Engineer',
                 11:'Information Security Specialist',
                 12:'Networking Engineer',
                 13:'Project Manager',
                 14:'Software Developer',
                 15:'Software Tester',
                 16:'Technical Writer'}

    #print(jobs_dict[predictions[0]])
    job = {}
    #job[0] = jobs_dict[predictions[0]]
    index = 1


    data1=predictions[0]
    print(data1)
    return
render_template("testafter.html",final_res=final_res,job_dict=jobs_dict,job0=data1)


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

Testmodel.py

```
import pandas as pd
import numpy as np
import pickle
career = pd.read_csv('dataset9000.data', header = None)
#np.dtype('float64')

X = np.array(career.iloc[:, 0:17]) #X is skills
print(X)
y = np.array(career.iloc[:, 17]) #Y is Roles
print("hi")
print(y)

##  attribute to return the column labels of the given Dataframe
career.columns = ["Database Fundamentals","Computer Architecture","Distributed
Computing Systems",
"Cyber Security","Networking","Development","Programming Skills","Project
Management",
"Computer Forensics Fundamentals","Technical Communication","AI ML","Software
Engineering","Business Analysis",
"Communication skills","Data Science","Troubleshooting skills","Graphics
Designing","Roles"]

career.dropna(how ='all', inplace = True)
#print("career.dropna(how ='all', inplace = True)",career.dropna(how ='all',
inplace = True))
career.head()
## splitting the data into training and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.3,
random_state = 524)
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
scores = {}
knn = KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train, y_train)
print('X_train',X_train)
print('y_train',y_train)
y_pred = knn.predict(X_test)
print('y_pred',y_pred)
scores[5] = metrics.accuracy_score(y_test, y_pred)
print('Accuracy=',scores[5]*100)

pickle.dump(knn, open('careerlast.pkl','wb'))
print('test file created')
```

bagging.py

```
from sklearn import model_selection
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
import numpy as np
import pickle
 # load the data
dataset = pd.read_csv('dataset9000.data', header = None)
print(dataset.head())
X=np.array(dataset.iloc[:, 0:17])
print(X)
Y = np.array(dataset.iloc[:, 17])
print(Y)
dataset.columns= ["Database Fundamentals","Computer Architecture","Distributed
Computing Systems",
"Cyber-Security","Networking","Development","Programming Skills","Project
Management",
"Computer Forensics Fundamentals","Technical Communication","AI ML","Software
Engineering","Business Analysis",
"Communication skills","Data Science","Troubleshooting-skills","Graphics
Designing","Roles"]
dataset.dropna(how ='all', inplace = True)

 seed =5
kfold = model_selection.KFold(n_splits = 15,
                      random_state = seed)
 # initialize the base classifier
base_cls = DecisionTreeClassifier()
 # no. of base classifier
num_trees = 50
 # bagging classifier
model = BaggingClassifier(base_estimator = base_cls,
                          n_estimators = num_trees,
                          random_state = seed)
 results = model_selection.cross_val_score(model, X, Y, cv = kfold)
print("accuracy :",results.mean()*100)
```

Hometest.html

```
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="utf-8">
       <meta http-equiv="X-UA-Compatible" content="IE=edge">
       <meta name="viewport" content="width=device-width, initial-scale=1">
        <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->

       <title>Intelligent Career Guidance System</title>

       <!-- Google font -->
       <link
href="https://fonts.googleapis.com/css?family=Lato:700%7CMontserrat:400,600"
rel="stylesheet">


 <link href="{{ url_for('static', filename='chart.css') }}" rel="stylesheet"
media="all">
  <link href="{{ url_for('static', filename='css1/bootstrap.min.css') }}"
rel="stylesheet" media="all">
  <link href="{{ url_for('static', filename='css1/font-awesome.min.css') }}"
rel="stylesheet" media="all">
  <link href="{{ url_for('static', filename='css1/s.css') }}" rel="stylesheet"
media="all">
 <link href='https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400'
rel='stylesheet' type='text/css'>
</head>
<body>
<!--Header-->
       <header id="header" class="transparent-nav" style="position:
fixed;background-color: rgb(40, 30, 241); top: 0;">
           <div class="container">

               <div class="navbar-header">
                   <!-- Logo -->
                   <div class="navbar-brand">
                       <a class="logo"
href="http://127.0.0.1/php%20CAREERLY/main.php">Career.io</a>
                   </div>
                   <!-- /Logo -->

                   <!-- Mobile toggle -->
                   <button class="navbar-toggle">
                       <span></span>
                   </button>
                   <!-- /Mobile toggle -->
               </div>
```

```
                <?php
                    // Check if the user is logged in, if not then redirect him to
login page
                    if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !==
true):
                ?>

                <!-- Navigation -->

                <?php else: ?>
                    <!-- Navigation -->
                <nav id="nav">
                    <ul class="main-menu nav navbar-nav navbar-right">
                        <li><a
href="http://127.0.0.1/php%20CAREERLY/main.php">Home</a></li>
                        <li class="dropdown">
                            <a class="dropbtn" href="javascript:void(0)">Services
<span>&#11167;</span></i></a>
                            <div class="dropdown-content">
                                <a
href="http://127.0.0.1/php%20CAREERLY/main.php">Career Prediction</a>
                                <a
href="http://127.0.0.1/php%20CAREERLY/courses.php">Courses</a>
                                <!--<a
href="http://127.0.0.1/php%20CAREERLY/blog.php">Blogs</a>
                                <a
href="http://127.0.0.1/php%20CAREERLY/daily_bytes.php">Daily bytes</a>
                                <a
href="http://127.0.0.1/php%20CAREERLY/community.php">Community</a>-->
                                <a
href="http://127.0.0.1/php%20CAREERLY/blog.php">Knowledge Network</a>

                            </div>
                        </li>
                        <li><a href="#about">About Us</a></li>
                        <li><a
href="http://127.0.0.1/php%20CAREERLY/contact.php">Contact Us</a></li>
                        <li><a href="http://127.0.0.1/php%20CAREERLY/logout.php"
>Log out</a></li>


                    </ul>
                </nav>
                <!-- /Navigation -->
                <?php endif ?>

            </div>
        </header>
        <div class="hero-area section">

            <!-- Backgound Image -->
```

```
            <div class="bg-image bg-parallax overlay"
style="background-image:url(static/img/bgc2.jpg); " ></div>
            <!-- /Backgound Image -->

            <div class="container">
                <div class="row" >
                    <div class="col-md-10 col-md-offset-1 text-center">
                        <ul class="hero-area-tree">
                            <li><a
href="http://127.0.0.1/php%20CAREERLY/main.php">Services</a></li>
                            <li><a href="http://127.0.0.1:5000/">Career
Prediction</a></li>
                        </ul>
                        <h1 class="white-text">Get Started to Discover
Yourself!</h1>

                    </div>
                </div>
            </div>

        </div>

    <!-- Icons font CSS-->

    <link href=" {{ url_for('static',
filename='vendor/mdi-font/css/material-design-iconic-font.min.css') }}"
rel="stylesheet" media="all">

    <link href="{{ url_for('static',
filename='vendor/font-awesome-4.7/css/font-awesome.min.css') }}" rel="stylesheet"
media="all">
    <link href="{{ url_for('static', filename='css/range.scss') }}" rel="stylesheet"
media="all">
    <!-- Font special for pages-->
    <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,
700,700i,800,800i" rel="stylesheet">

    <!-- Vendor CSS-->

    <link href="{{ url_for('static', filename='vendor/select2/select2.min.css') }}"
rel="stylesheet" media="all">

    <link href="{{ url_for('static',
filename='vendor/datepicker/daterangepicker.css') }}" rel="stylesheet" media="all">
      <!-- Main CSS-->

    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet"
media="all">
</head>
```

```
<body>
   <div class="page-wrapper bg-gra-03 p-t-45 p-b-50">
       <div class="wrapper wrapper--w790">
           <div class="card card-5">
               <div class="card-heading">
                   <h2 class="title">Rate Yourself.....!!</h2>

               </div>


               <div class="card-body">

                   <form  method="POST", action="{{url_for('result')}}">

                       <div class="form-group row">
                           <label  class="col-sm-3  col-form-label">Database
Fundamentals</label>
                           <div class="col-sm-9">
                           <select class="form-select form-control" required
aria-label="select example" name="rate_Database Fundamentals">
                                 <option value="">Choose option</option>
                                <option value="1">Not Interested</option>
                                   <option value="2">Poor</option>
                                   <option value="3">Beginner</option>
                                   <option value="5">Average</option>
                                   <option value="6">Intermediate</option>
                                   <option value="7">Excellent</option>
                                   <option value="9">Professional</option>
                               </select>

                           </div>
               </div>
               <div class="form-group row">
                           <label  class="col-sm-3  col-form-label">Computer
Architecture</label>
                           <div class="col-sm-9">
                            <select class="form-select form-control" required
aria-label="select example" name="rate_Computer Architecture" >
                                   <option value="">Choose option</option>
                                   <option value="1">Not Interested</option>
                                   <option value="2">Poor</option>
                                   <option value="3">Beginner</option>
                                   <option value="5">Average</option>
                                   <option value="6">Intermediate</option>
                                   <option value="7">Excellent</option>
                                   <option value="9">Professional</option>
                            </select>
                           </div>
               </div>
               <div class="form-group row">
               <div class="form-group row">
```

```
                        <label  class="col-sm-3 col-form-label">Data
Science</label>
                        <div class="col-sm-9">
                         <select class="form-select form-control" required
aria-label="select example" name="rate_Data Science" >
                                <option value="">Choose option</option>
                                <option value="1">Not Interested</option>
                                <option value="2">Poor</option>
                                <option value="3">Beginner</option>
                                <option value="5">Average</option>
                                <option value="6">Intermediate</option>
                                <option value="7">Excellent</option>
                                <option value="9">Professional</option>
                         </select>
                        </div>
            </div>
            <div class="form-group row">
                        <label  class="col-sm-3  col-form-label">Troubleshooting
skills</label>
                        <div class="col-sm-9">
                         <select  class="form-control"
name="rate_Troubleshooting skills" >
                                <option value="">Choose option</option>
                                <option value="1">Not Interested</option>
                                <option value="2">Poor</option>
                                <option value="3">Beginner</option>
                                <option value="5">Average</option>
                                <option value="6">Intermediate</option>
                                <option value="7">Excellent</option>
                                <option value="9">Professional</option>
                         </select>
                        </div>

            </div>
            <div class="form-group row">
                        <label  class="col-sm-3  col-form-label">Graphics
Designing</label>
                        <div class="col-sm-9">
                         <select  class="form-control" name="rate" >
                                <option value="">Choose option</option>
                                <option value="1">Not Interested</option>
                                <option value="2">Poor</option>
                                <option value="3">Beginner</option>
                                <option value="5">Average</option>
                                <option value="6">Intermediate</option>
                                <option value="7">Excellent</option>
                                <option value="9">Professional</option>
                         </select>
                        </div>

            </div>
```

```
                <center><button class="btn btn--radius-2 btn--red" type="submit"
>Discover Yourself!</button></center>
                </div>
                </div>
</div>

</form>

   <!-- Footer -->
       <footer id="footer" class="section">

           <!-- container -->
           <div class="container">

               <!-- row -->
               <div class="row">

                   <!-- footer logo -->
                   <div class="col-md-6">
                       <div class="footer-logo">
                           <a class="logo" style="font-size: 30px;"
href="main.php">Career.io</a>
                       </div>
                   </div>
                   <!-- footer logo -->



               </div>
               <!-- /row -->

               <!-- row -->
               <div id="bottom-footer" class="row">

                   <!-- social -->
                   <div class="col-md-4 col-md-push-8">

                   </div>
                   <!-- /social -->

                   <!-- copyright -->
                   <div class="col-md-8 col-md-pull-4">
                       <div class="footer-copyright">
                           <span>&copy; Copyright 2021. All Rights Reserved.
</span>
                       </div>
                   </div>
                   <!-- /copyright -->
                   </div>
               <!-- row -->

           </div>
```

```
            <!-- /container -->

        </footer>
        <!-- /Footer -->


</body>

</html>
```