INSIGHTIFY

PROJECT REPORT

21AD1513- INNOVATION PRACTICES LAB

Submitted by

ABISHEK.S	**211422243012**
DHARANI KUMAR.P.Y	**211422243060**
DILIP KUMAR S	**211422243068**

in partial fulfillment of the requirements for the award of degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123

ANNA UNIVERSITY: CHENNAI-600 025

NOVEMBER, 2024

BONAFIDE CERTIFICATE

Certified that this project report titled **"INSIGHTIFY"** is the bonafide work of **ABISHEK.S (211422243012), DHARANI KUMAR.P.Y (211422243060), S DILIP KUMAR (211422243060)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

INTERNAL GUIDE
Mrs. V.Rekha M.E.,
Assistant Professor
Department of AI &DS

HEAD OF THE DEPARTMENT
Dr.S.MALATHI M.E., Ph.D
Professor and Head,
Department of AI & DS.

Certified that the candidate was examined in the Viva-Voce Examination held on ……

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

This report introduces Insightify, an online platform designed to support users in their educational and career development journeys. Insightify provides personalized tools for Course Recommendations, Roadmap Generation, Resume Enhancement, Project Recommendations, and a Quiz Module. These features work collaboratively to simplify skill-building and career planning, offering tailored guidance for each user. Through a user-friendly interface, Insightify enables users to explore learning pathways, strengthen resumes, and assess progress, effectively empowering them in achieving their professional goals.

Keywords : *Insightify, Educational platform, Career development, Personalized tools, Course recommendations, Roadmap generation, Resume enhancement, Project recommendations, Quiz module, Skill-building, Career planning, User-friendly interface, Learning pathways, Professional goals, Tailored guidance.*

ACKNOWLEDGEMENT

I also take this opportunity to thank all the Faculty and Non-Teaching Staff Members of Department of Computer Science and Engineering for their constant support. Finally I thank each and every one who helped me to complete this project. At the outset we would like to express our gratitude to our beloved respected Chairman, **Dr.Jeppiaar M.A.,Ph.D,** Our beloved correspondent and Secretary **Mr.P.Chinnadurai M.A., M.Phil., Ph.D.,** and our esteemed director for their support.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr.S.Malathi M,E.,Ph.D.,** of Artificial Intelligence and Data Science for her encouragement.

Personally we thank **V.REKHA,M.E Assistant Professor ,**Department of Artificial Intelligence and Data Science for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our thanks to the project coordinators **V.REKHA,M.E Assistant Professor** in Department of Artificial Intelligence and Data Science for their Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, friends, and well-wishers who have helped us for the successful completion of our project.

We also take the opportunity to thank all faculty and non-teaching staff members in our department for their timely guidance in completing our project.

ABISHEK.S	**DHARANI KUMAR.P.Y**	**DILIP KUMAR S**
(211422243012)	**(211422243060)**	**(211422243068)**

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATIONS	MEANING
TF	TERMED FREQUENCY
IDF	INVERSE DOCUMENT FREQUENCY
API	APPLICATION PROGRAMMING INTERFACE

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1. INTRODUCTION

1.1 Introduction To Insightify

In the modern landscape of education and career advancement, individuals face an array of choices and challenges as they strive to improve their skills and enhance their employability. The advent of technology has transformed the way education is delivered, with online learning platforms offering unprecedented access to a vast pool of knowledge and resources. However, despite these advancements, many learners and job seekers still struggle to navigate the complexities of skill acquisition and career development effectively. Insightify emerges as a comprehensive solution designed to support users in their educational and professional journeys by providing personalized tools and resources tailored to their specific needs.

The digital age has ushered in a shift in the workforce, with employers increasingly seeking candidates who possess not only formal qualifications but also relevant skills and practical experience. As industries evolve, the demand for continuous learning and upskilling has never been greater. Individuals are now expected to take charge of their own educational pathways, but the sheer volume of available information can be overwhelming. Students and professionals often find themselves at a crossroads, unsure of which courses to pursue, what skills to develop, or how to effectively present themselves to potential employers. This scenario is compounded by the fact that many existing educational platforms lack the necessary features to provide cohesive support, leading to frustration and disengagement.

1.2 Challenges Faced In Educational And Career Development

Despite the proliferation of online learning platforms, several challenges persist that hinder effective educational and career development. These challenges include:

1. **Overwhelming Course Options:** With countless online courses available across various platforms, users often experience analysis paralysis when trying to select the right ones. This overwhelming abundance of information can lead to decision fatigue, resulting in inaction and missed opportunities**.**
2. **Lack of Personalized Guidance:** Many educational platforms employ a one-size-fits-all approach, offering generic recommendations that fail to consider individual learning styles, preferences, and career objectives. Without tailored guidance, users may struggle to engage fully with the material, leading to lower retention and motivation.

3. **Ineffective Resume Building:** Crafting a compelling resume is a critical step in the job search process. However, many users lack the knowledge or resources to create documents that effectively showcase their strengths. A poorly constructed resume can significantly reduce a candidate's chances of securing interviews and job offers.
4. **Fragmented Learning Experience**: Users often find themselves juggling multiple platforms and tools for different aspects of their educational journey, such as skill assessments, project recommendations, and progress tracking. This fragmentation not only complicates the learning process but also diminishes the overall user experience.
5. **Limited Assessment Tools**: Many platforms do not provide adequate assessments or quizzes to help users evaluate their understanding and mastery of course content. Without effective self-assessment tools, learners may be unaware of their progress and areas needing improvement.

Insightify addresses these challenges head-on by integrating essential educational and career development tools into a single platform. By providing a seamless and cohesive user experience, Insightify empowers individuals to take charge of their learning and career trajectories, ensuring they are well-prepared for the demands of today's job market.

1.3 Objectives Of The Platform

The primary objective of Insightify is to create an integrated, user-friendly platform that enhances the educational and career development journey for users. By providing tailored resources and tools, the platform aims to support individuals in achieving their learning and professional goals. The specific objectives of Insightify are as follows:

1. **Personalized Course recommendations:** Offer users tailored course suggestions based on their interests, skills, and career aspirations, utilizing advanced algorithms for enhanced relevance and engagement. This ensures users can easily navigate the multitude of available courses and select those that align with their specific objectives.
2. **Roadmap Generation for Skill Development**: Create detailed learning roadmaps that guide users through their educational journey, linking relevant courses and resources to specific skills. This structured approach helps users visualize their progression and remain focused on their learning goals.
3. **Resume Enhancement Tools:** Provide users with tools to build and refine their resumes by matching their skills and experiences with job descriptions. This feature enables users to present themselves effectively to potential employers, increasing their chances of landing desired job opportunities.

4. **Project Recommendations for Practical Experience:** Suggest relevant projects that users can undertake to apply their skills in real-world scenarios, complete with descriptions, difficulty levels, and required tools. This hands-on experience is crucial for building a robust portfolio and demonstrating competency to prospective employers.
5. **Domain-Specific Quiz Module:** Implement a quiz module that offers domain-specific quizzes to assess users' knowledge and readiness in their chosen fields. This feature encourages self-assessment and reinforces learning, helping users identify areas for improvement.

1.4 Architecture Diagram

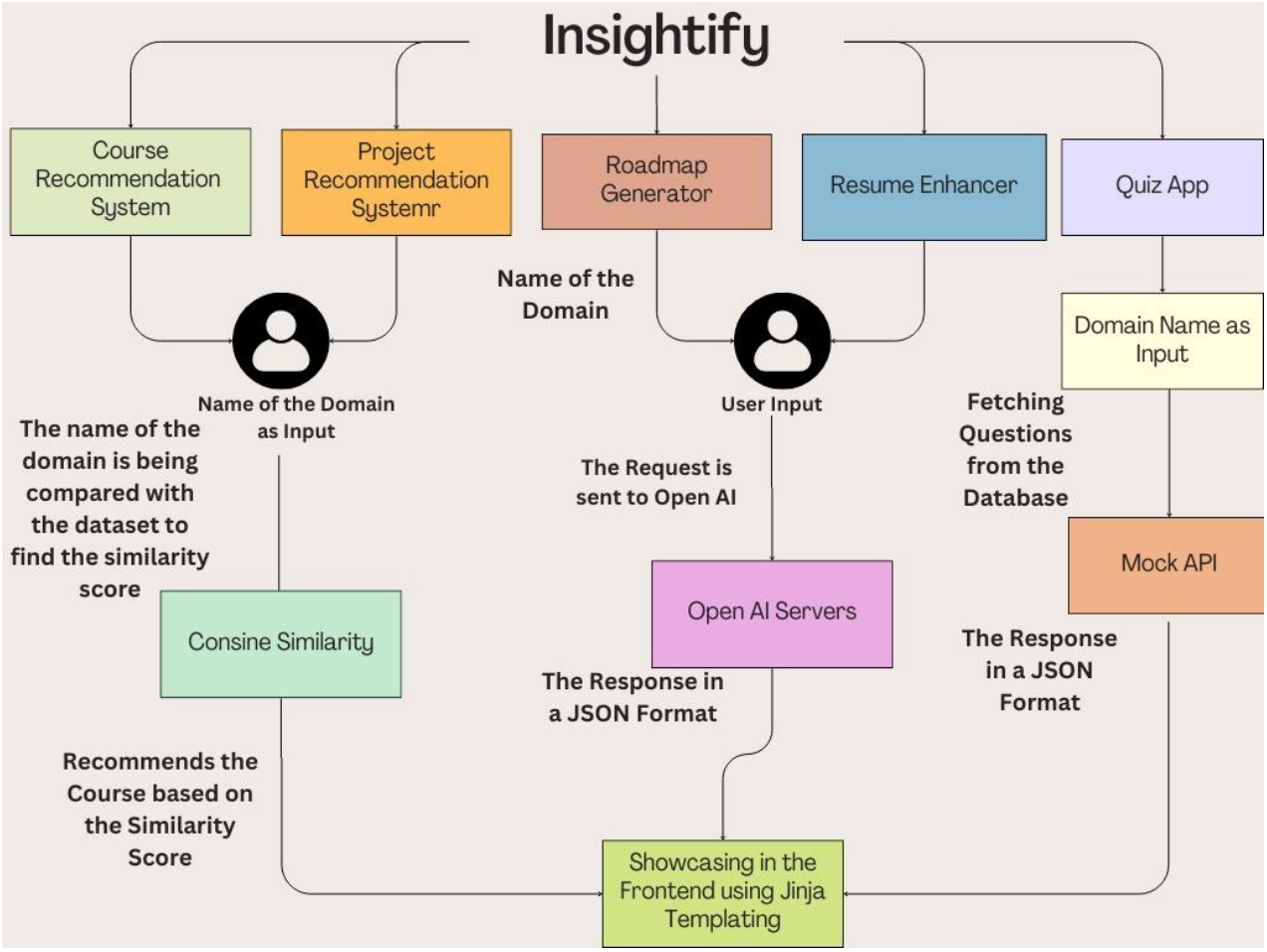

Fig 1.1 Architecture Diagram

The Insightify platform offers several key features designed to support student development through a modular architecture. At the core are tools such as the Course Recommendation System, Project Recommendation System, Roadmap Generator, Resume Enhancer, and Quiz App, each tailored to specific aspects of student learning and skill enhancement. The user begins by entering the domain name, which is processed by different modules based on the type of recommendation or information

requested. For the Course Recommendation System, cosine similarity is used to match the input domain with the dataset to identify and recommend relevant courses. The Project Recommendation System uses similar methods to suggest projects suited to the domain. For the Roadmap Generator and Resume Enhancer, user inputs are sent to OpenAI servers, which process the data and return JSON-formatted responses, enabling further customization based on student needs. The Quiz App pulls questions from a database via a mock API and provides JSON-formatted responses for seamless integration. Finally, the entire output is displayed in the frontend, powered by Jinja templating, allowing for dynamic content presentation. This architecture enables Insightify to provide personalized resources and guidance to students in an efficient and structured manner.

CHAPTER 2

2. LITERATURE REVIEW

The landscape of education and career development is evolving rapidly due to advancements in technology, particularly with the integration of artificial intelligence (AI) and data analytics. AI-driven educational platforms are transforming how learners access information, engage with materials, and develop the skills necessary for success in the workforce. Research indicates that personalized learning experiences significantly enhance learner engagement and satisfaction, which is crucial in an era where learners have diverse needs and goals.

Personalized course recommendation systems, powered by algorithms that analyze user data and preferences, are at the forefront of this transformation. Such systems utilize techniques like collaborative filtering and content-based filtering to suggest relevant courses, helping users navigate the overwhelming number of available options and ensuring that their learning pathways align with their career aspirations. By tailoring educational experiences to individual needs, these platforms foster a more engaging and effective learning environment.

Moreover, the use of AI in resume enhancement tools is gaining traction. These tools can analyze job descriptions and match them with users' qualifications, providing real-time feedback on how well their resumes align with desired positions. This not only helps job seekers present themselves more effectively but also increases their chances of securing interviews in a competitive job market. Research highlights that a well-optimized resume can significantly influence hiring decisions, emphasizing the importance of such tools in modern job searches.

Project-based learning, facilitated by AI-driven project recommendation systems, is another critical area in educational development. By suggesting projects based on users' skills and domains, these platforms provide opportunities for practical application of knowledge, which is essential for skill mastery and portfolio

development . Studies have shown that hands-on experience is a key factor in successful job placement, as employers increasingly value practical skills alongside formal education.

Additionally, the integration of assessment tools, such as quizzes and self-evaluation modules, enhances the learning process by enabling learners to track their progress and identify areas for improvement. Research supports the notion that regular assessments and feedback significantly boost retention rates and learner motivation [7]. Platforms that incorporate these features promote self-directed learning, empowering users to take ownership of their educational journeys [8].

However, the implementation of AI-driven educational tools is not without challenges. Concerns regarding data privacy and security remain paramount, especially as platforms collect and analyze sensitive user information [9]. Ensuring that users feel confident in the security of their data is crucial for fostering trust in these systems. Additionally, there is ongoing discourse about the potential biases embedded in AI algorithms, which could inadvertently affect the fairness of course recommendations and assessments [10].

Furthermore, the shift towards AI-based educational platforms raises questions about the evolving roles of educators and mentors in the learning process. While AI can enhance educational experiences, it cannot replace the essential human element of teaching and guidance [11]. Therefore, a balanced approach that integrates technology while preserving the critical interpersonal aspects of education is necessary for maximizing learner outcomes.

In conclusion, the integration of AI in educational platforms offers substantial benefits in terms of personalized learning, career readiness, and skill development. By leveraging technology to enhance user experiences, Insightify aims to create a holistic educational environment that supports learners in achieving their career goals. However, it is essential to address the associated challenges, particularly concerning data privacy and the ethical implications of AI, to ensure that technology serves to empower users rather than hinder their progress [12].

CHAPTER 3

3. COURSE RECOMMENDER

3.1. Introduction To Course Recommendation Systems

As the demand for online learning platforms increases, students and professionals face challenges in selecting the most appropriate courses that match their interests and learning goals. Recommender systems serve a crucial role in addressing this issue by filtering and suggesting relevant courses. This paper presents a recommendation system that leverages data from platforms like Udemy, Coursera, and edX to offer personalized course suggestions based on course titles and content descriptions.

3.2. Data Sources

The project utilizes course data scraped from three widely recognized e-learning platforms: Udemy, Coursera, and edX. These platforms collectively offer a diverse range of courses across various domains, including technology, business, and personal development. Below are the details of each dataset:

3.2.1. Udemy Dataset

The Udemy dataset consists of over 40,000 courses spanning numerous categories such as programming, design, marketing, and more. Each course in the dataset contains attributes such as:

- Title: The name of the course.
- Description: A brief overview of the course content.
- Rating: An average rating based on user reviews.
- Number of Reviews: The total count of student reviews.
- Price: The cost of enrolling in the course.

The dataset was scraped from Udemy's public catalogue and made available through Kaggle. The dataset was sourced from the Udemy catalogue and made available on Kaggle [1].

3.2.2. COURSERA Dataset

Coursera offers a large collection of university and industry-affiliated courses, with data covering both free and paid learning options. The dataset contains information such as:

- Title: The official title of the course.
- Institution: The university or organization offering the course.
- Description: A brief summary of the course objectives and content.
- Rating: An aggregate rating based on student feedback.
- Enrollment Count: The number of students who have enrolled in the course.

This dataset includes professional certifications and specializations that extend beyond individual courses. The Coursera dataset was retrieved from Kaggle [2].

3.2.3. edX Dataset

edX, a platform providing courses from top-tier universities such as MIT and Harvard, offers open courses across a variety of fields, from computer science to humanities. The dataset includes key course attributes such as:

- Title: The course's official name.
- Institution: The university or organization offering the course.
- Subject: The academic discipline or topic of the course.
- Description: A short description of the course curriculum.
- Level: The difficulty level (beginner, intermediate, advanced) of the course.

The edX dataset can be accessed via Kaggle [3].

3.2.4. Data Preprocessing

Before applying any recommendation techniques, the datasets were pre-processed to ensure consistency across all platforms. This included:

- **Text Normalization**: Lowercasing all course titles and descriptions to ensure uniformity.
- **Handling Missing Values**: Removing courses that lacked sufficient information, such as missing descriptions or ratings.
- **Punctuation and Special Characters Removal**: Stripping punctuation and special symbols from textual data to avoid noise during similarity calculations.

- **Rating Normalization**: Converting course ratings to a common scale where necessary, and filling in missing values using imputation techniques.

This cleaned and structured data serves as the foundation for the recommendation system, ensuring that courses are appropriately compared across platforms.

3.3. Natural Language Processing Techniques

To efficiently process and compare textual course data, natural language processing (NLP) techniques were employed. Specifically, the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method was used to convert textual data (course titles and descriptions) into numerical feature vectors.

3.3.1. TF-IDF Vectorization And Feature Vector Creation

TF-IDF is a statistical measure used to evaluate how important a word is to a document relative to a collection of documents. In this context, it helps quantify the importance of words within course titles and descriptions from each platform**.**

3.3.1.1. Term Frequency (TF)

Refers to the number of times a word appears in a given document (in this case, a course title or description). For instance, in a course description for "Data Science," terms like "data" or "science" may appear multiple times, and their TF score increases accordingly.

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

3.3.1.2. Inverse Document Frequency (IDF)

The inverse frequency of the term across all documents. This helps downplay the importance of terms that appear frequently across many courses (e.g., "course," "learning") and highlights more unique terms that can distinguish a course from others.

$$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

N is the total number of documents (course descriptions).

df(t) is the number of documents containing the term t.

3.3.2. Combined Effect Of TF-IDF

By combining both TF and IDF, the system was able to generate a weighted score that reflects the importance of each word in relation to the entire dataset. Terms that are common in a particular course but rare across all other courses (e.g., "blockchain" in a Blockchain course) receive higher scores, allowing the system to identify and emphasize key concepts that differentiate courses.

3.2.2.1. Platform-Specific TF-IDF Vectorizers

Each e-learning platform (Udemy, Coursera, and edX) offered different sets of courses with unique structures and content, so separate **TF-IDF vectorizers** were trained for each platform. This ensured that the specific vocabulary and key terms from each platform were captured accurately in the recommendation process.

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

- **Udemy Vectorizer**: The Udemy dataset was first preprocessed by removing stop words, punctuation, and other noise. The course titles and descriptions were then vectorized using TF-IDF, creating a matrix of numerical vectors where each row represented a course, and each column represented a term's importance score.
- **Coursera Vectorizer**: Similarly, the Coursera dataset was processed and vectorized using TF-IDF to capture the nuances of university-affiliated and professional courses. The vectorization captured the terms specific to Coursera's structured content (e.g., specializations, professional certificates).
- **edX Vectorizer**: For edX, which primarily focuses on higher education courses from top universities, the TF-IDF vectorizer highlighted key academic terms that distinguish the platform's offerings. This allowed the system to represent edX courses as vectors with important terms receiving higher weights.

These separate vectorizers ensured that the system accurately processed and compared courses within each platform.

Create an image of TF IDF and Feature Vector

Remember Do a common heading for Data Source and Performance Metrices

3.4. Cosine Similarity For Course Recommendations

After vectorizing the course data using TF-IDF, the next step involved comparing these vectors to recommend courses that were most similar to a user's search query. Cosine similarity was the primary technique used for this purpose.

3.4.1 Introduction To Cosine Similarity

Cosine similarity is a measure used to calculate the similarity between two non-zero vectors by computing the cosine of the angle between them. In the context of this project, the vectors represent the TF-IDF-transformed course descriptions. Cosine similarity measures how close these vectors are to one another, with values ranging from:

- 1 (indicating that the vectors are identical),
- 0 (indicating no similarity),
- -1 (indicating that the vectors are diametrically opposed).

The formula for cosine similarity is given as:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:

- A and B are the TF-IDF vectors of two courses,
- $A \cdot B$ is the dot product of the vectors,
- $||A||$and $||B||$ are the magnitudes of the vectors.

By comparing vectors, cosine similarity enabled the recommendation system to identify courses with similar content based on their descriptions, even if they didn't share identical wording.

3.4.2 Application In The Project

The project utilized cosine similarity to compute the relevance of a course based on user input. Once a user provided a search query (e.g., "machine learning"), the query was transformed into a TF-IDF vector. This vector was then compared to the vectors of all courses within a platform (Udemy, Coursera, or edX) to find the ones with the highest similarity scores.

Here's how the recommendation process worked:

1. **Vectorizing the Input Query**: The user's search query (e.g., "Data Science") was first converted into a TF-IDF vector using the pre-trained vectorizer for the platform.
2. **Computing Similarity**: Cosine similarity was applied to measure the similarity between the vectorized query and the vectors of all available courses within the

dataset. Courses that shared a high degree of similarity (i.e., had a cosine similarity score close to 1) were identified as top matches.

3. **Sorting and Filtering**: The similarity scores were sorted in descending order, with the highest scores representing the most relevant courses. Filters were applied to remove courses that did not meet additional criteria, such as minimum ratings.
4. **User Interface Presentation:** The final list of recommended courses is displayed to the user through the application interface. Each recommendation typically includes:

    a. Course title
    b. Description
    c. Rating
    d. Platform name
    e. Any other relevant information (e.g., instructor name, course duration).

For instance, when a user searched for "machine learning," the system transformed this query into a TF-IDF vector and compared it with the vectors of courses from Udemy, Coursera, and edX. The system then recommended the courses with the highest cosine similarity to the query, providing results that were highly relevant to the user's interests.

CHAPTER 4

4. ROAD MAPPER

4.1 Introduction To Road Mapper

Road Mapper is an intelligent tool designed to generate personalized roadmaps based on specific domains provided by the user. By analyzing the requirements, trends, and key learning milestones in a given field, Road Mapper outlines a clear, structured pathway for users to follow, ensuring they acquire the necessary knowledge and skills efficiently using Open AI models. This feature is particularly useful for individuals looking to streamline their learning journey and stay competitive in their chosen domain.

4.2 Working Of Road Mapper

4.2.1 User Input Query

The Open AI model requires a user-defined input for the domain parameter, which represents the specific subject or field of interest. This input can be gathered through

various means, such as a command-line prompt or a graphical user interface. For example, if the user intends to generate a roadmap for "Data Science,"

4.2.2 Request Structure Using Open Ai Models

initiates a request to OpenAI's API, specifically using the *ChatCompletion* endpoint. The objective is to generate a JSON roadmap for educational resources across five proficiency levels: Beginner, Intermediate, Advanced, Expert, and Specialized. The structure of the request consists of two main components: system messages and user prompts.

1. **System Message**

```
{"role": "system", "content": "You are a helpful assistant."}
```

- Defines the assistant's role and context:

2. **User Message**

```
{"role": "user",
 "content": f"""Generate a JSON roadmap for the domain of{domain} with five
levels:Beginner, Intermediate, Advanced, Expert, and Specialized. Each level
should include a description and at least 6 curated resources with titles and
URLs.}
```

- Contains the detailed instructions for generating the roadmap:

4.2.3 Sending The Request

Once the user input has been gathered and the messages are structured, the request is sent to OpenAI's API using the following method:

```
response = openai.ChatCompletion.create(
   model = "gpt-3.5-turbo",
   messages = messages,
   max_tokens = 1024,
   n = 1,
   temperature = 0.6,
)
```

- **model**: Specifies the model to be used for generating responses (e.g., "gpt-3.5-turbo").
- **messages**: Contains the structured conversation history, including both system and user messages.
- **max_tokens**: Sets the maximum number of tokens in the response generated by the model (e.g., 1024).
- **n**: Indicates the number of response completions to generate (e.g., 1).

- **temperature**: Controls the randomness of the output; higher values produce more varied responses (e.g., 0.6).

4.2.4 Structure Of Output

The Road Mapper provides results of syllabus in five distinct levels:

4.2.4.1. Beginner

The Beginner level provides a foundational starting point for users new to the domain. The courses focus on building basic knowledge. For example, if the selected domain is **Data Science**, the system might recommend:

- Data Science for Beginners
- Python for Data Science
- Introduction to Machine Learning
- Data Visualization Basics
- Statistics for Data Science

4.2.4.2. Intermediate

At the **Intermediate** level, the courses delve deeper into the domain, helping users gain a more thorough understanding. For the **Data Science** domain, some intermediate-level courses might include:

- Machine Learning: Regression
- Data Cleaning and Preprocessing
- Deep Learning Specialization
- Feature Engineering for Machine Learning
- Natural Language Processing

4.2.4.3. Advanced

The **Advanced** level is for users with significant experience in the domain. For **Data Science**, advanced-level course recommendations might include:

- Advanced Machine Learning Algorithms
- Deep Learning with TensorFlow
- Reinforcement Learning Fundamentals
- Advanced Statistical Modeling
- Natural Language Generation

4.2.4.4. Expert

The **Expert** level focuses on highly advanced, professional-grade knowledge. These courses help users become authorities in the domain. For **Data Science**, some expert-level recommendations could include:

- AI Model Optimization
- Scalable Machine Learning Systems
- Big Data Analytics
- Real-time Data Processing
- High-Performance Computing for Machine Learning

4.2.4.5. Specialized

The **Specialized** level is designed for users looking to explore niche areas or emerging topics within the domain. For instance, in **Data Science**, specialized courses might cover:

- Quantum Machine Learning
- AI for Healthcare

- Blockchain for Data Science
- Edge Computing for AI
- Autonomous Systems Development

CHAPTER 5

5. PROJECT RECOMMENDER

The **Project Recommender** system is a user-friendly platform designed to suggest projects based on a user's chosen domain and difficulty level. The system streamlines the process of project selection by providing recommendations that align with the user's skill set and interests.

5.1 User Inputs

The system takes two key inputs from the user:

1. **Domain Selection:** Users can choose a domain that they are interested in (e.g., Artificial Intelligence, Web Development, Cybersecurity, etc.).
2. **Difficulty Level:** Users select a difficulty level from the following:
    - **Easy**
    - **Intermediate**
    - **Advanced**

5.2 Output Structure Of Project Recommendations

Once the user inputs their preferences, the system generates and displays a list of recommended projects that match the specified domain and difficulty level. Each recommendation provides the following information:

- **Project Title:** A descriptive title of the project.
- **Difficulty Level:** The complexity of the project.
- **Estimated Time:** The time required to complete the project.
- **Tools/Technologies:** Technologies or tools necessary to complete the project.
- **Prerequisites:** Knowledge or skills required before starting the project.

- **Learning Outcomes:** Key skills or knowledge that will be gained upon completing the project.

Example Output:

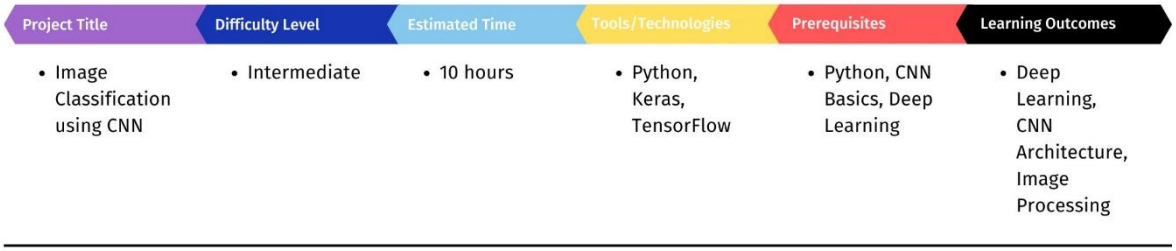

Fig 1.2 Difficulty Level

5.3 Dataset Structure

The system is powered by a custom-built dataset that links project titles with domains and difficulty levels. This dataset is a collection of project metadata, including:

- **Domain:** The specific field or area the project belongs to (e.g., Data Science, Web Development).
- **Difficulty Level:** Classifies the projects as Easy, Intermediate, or Advanced.
- **Project Metadata:** Each project entry includes:
    - Project Title
    - Estimated Completion Time
    - Tools and Technologies Required
    - Prerequisites for the Project
    - Learning Outcomes

The dataset is stored in a structured format, such as a CSV file or a relational database, and is used to dynamically fetch project suggestions based on user input.

5.4 Recommendation Algorithm

The core of our project recommender system is the recommendation algorithm, designed to filter and suggest projects based on the user's input regarding the domain of interest and the desired difficulty level. This algorithm processes the dataset containing project information, applies filtering conditions, and returns a curated list of project suggestions.

The recommendation algorithm follows a structured approach:

Step 1: Dataset Loading

The algorithm begins by loading a dataset that contains a comprehensive list of projects. This dataset includes key attributes such as project title, domain, difficulty level, estimated time to complete, required tools, prerequisites, and learning outcomes.

Step 2: Filtering Based on User Input

Once the dataset is loaded, the user selects their preferred domain and difficulty level. The algorithm then filters the dataset to retain only those projects that match both the selected domain and difficulty level.

Step 3: Recommendation Generation

After filtering, the algorithm presents the user with a list of projects that align with their criteria. Each recommended project includes detailed information such as the project title, difficulty level, required tools, prerequisites, estimated time for completion, and the key learning outcomes.

This structured filtering process ensures that the user receives personalized project recommendations that match their skill level and area of interest.

```
import pandas as pd

# Load project data
def load_data(file_path):
    return pd.read_csv(file_path)

# Filter projects by domain and difficulty
def filter_projects(data, domain, difficulty):
    return data[(data['domain'] == domain) & (data['difficulty'] == difficulty)]

# Display recommendations
def show_recommendations(filtered_data):
    if filtered_data.empty:
        print("No projects found.")
    else:
        for _, row in filtered_data.iterrows():
            print(f"Project: {row['project_title']}, Difficulty: {row['difficulty']}")
```

5.5 Future Enhancements With Openai Api

To further enhance the system's capabilities, especially for providing real-time and up-to-date project recommendations across many dynamically evolving domains, the OpenAI API can be integrated. This will enable the Project Recommender to:

- Fetch and generate new project ideas based on real-time domain-specific advancements.
- Tap into an expanded knowledge base of continuously updated technical trends and innovative project suggestions from cutting-edge fields like Quantum Computing, Blockchain, or Advanced Robotics.

By integrating the OpenAI API, the system can adapt to the rapidly changing technology landscape, offering users an ever-growing range of project recommendations based on real-world trends and insights.

5.6 User Workflow

1. The user selects a domain and a difficulty level from a drop-down menu.
2. The system queries the dataset based on these inputs.
3. Relevant project recommendations are displayed on the screen in a structured format, with details such as project title, estimated completion time, tools, prerequisites, and learning outcomes.

By integrating AI-powered content generation and real-time domain insights, the Project Recommender ensures users receive relevant, personalized project suggestions tailored to their preferences and skill levels. This system also holds the potential for continuous improvement through OpenAI API integration, which can keep the recommendations aligned with the latest trends in various technological domains.

CHAPTER 6

6. RESUME ENHANCER

6.1 Overview

The Resume Enhancer is a sophisticated tool designed to assist job candidates in optimizing their resumes to better align with specific job descriptions. By employing Natural Language Processing (NLP) techniques, the tool evaluates the textual similarities between a user's resume and a job listing, providing a quantifiable match percentage. This percentage serves as an indicator of how well the resume meets the requirements outlined in the job description, helping candidates identify areas for improvement and increasing their chances of securing an interview.

6.2 User Input Query

6.2.1 Resume As Input

A comprehensive document that outlines the candidate’s skills, work experiences, qualifications, and achievements. This document may vary in format and content based on individual preferences and career stages. And highly the document is expected in the form of .pdf(portable document file)

6.2.2 Job Description As Input

The textual content of a job listing, which specifies the qualifications, skills, responsibilities, and experience required for a particular position. Job descriptions

often include both mandatory and preferred qualifications, making it essential for candidates to understand the nuances of the role.

Below is an example of a Job Description for a Full-Stack Developer

Job Summary:

We are looking for a talented Full Stack Developer to design, develop, and maintain web applications. You'll work on both front-end interfaces and back-end systems, ensuring a seamless user experience. The ideal candidate should be proficient in modern frameworks, databases, and agile development practices.

Key Responsibilities:

- *Develop responsive UIs using HTML, CSS, and JavaScript frameworks (React, Angular, Vue.js).*
- *Build back-end services and APIs using Node.js, Python, or similar technologies.*
- *Manage databases (SQL/NoSQL) for efficient data handling.*
- *Perform testing, debugging, and optimize applications for performance.*
- *Collaborate with cross-functional teams in an Agile environment.*

Qualifications:

- *Proficiency in front-end and back-end development (e.g., React, Node.js, Python).*
- *Experience with databases (MySQL, PostgreSQL, MongoDB).*
- *Strong problem-solving skills and ability to work in a fast-paced environment.*

6.3 Resume Score Enhancer Using Cosine Similiarity

The **Resume Score** is calculated by comparing the user's resume with a given job description. The system uses **cosine similarity refer(1.3.1)** to compute the alignment between the skills, experiences, and qualifications listed in the resume and the required job description. This similarity score is a key metric that determines how well a user's resume matches specific job roles.

- **Example**: A resume with data science experience will be matched with a job description for a "Data Scientist," resulting in a score. If the experiences and skills align closely, the similarity score will be higher (e.g., 85%). In contrast, a mismatch (e.g., applying for a "Web Designer" role with no related experience) will yield a lower score.

The **Resume Score** is dynamically generated and displayed to users to indicate how competitive their resumes are for specific job roles.

6.4 Application Of Open Ai In Resume Enhancer

6.4.1 Jobs You Can Apply For

This section displays a list of jobs the user is eligible to apply for based on the content of their resume. The OpenAI GPT model processes the resume text, extracts key qualifications, skills, and experiences, and uses this information to suggest job titles that match the profile (For a detailed explanation of **Working of Open AI**, refer to Section **2.2 Working of Road Mapper**. The same principle has been applied here). And Based on a resume that lists proficiency in Python, machine learning, and data analytics, the model might suggest the following jobs:

- **Data Analyst** at a consulting firm
- **Machine Learning Engineer** at a tech company
- **Data Science Intern** at a research institute
- **Business Intelligence Analyst** at a financial services firm
- **AI Developer** at a software development company

In this process, the OpenAI API takes the resume as input, processes it, and generates relevant job titles by comparing the skills listed in the resume with job roles available in the market. The role of the OpenAI API here is essential for understanding the user's profile and generating tailored job suggestions.

6.4.2 Pros And Cons Of The Resume

The **Pros and Cons** section provides a personalized analysis of the resume. Using the GPT model, the system identifies five strong points that make the resume attractive to potential employers and five areas where the resume could be improved. This feedback is invaluable for users looking to refine and optimize their resumes.

- **Pros**: These typically highlight key strengths such as relevant experience, technical skills, leadership roles, or certifications that align with job requirements.
- **Cons**: These might point out areas for improvement, such as gaps in employment, lack of specific technical expertise, or unclear role descriptions.

The OpenAI API plays a significant role here, taking in the resume text and using its language understanding capabilities to generate a structured response that highlights

both strengths and weaknesses. This feedback helps users tailor their resumes for better job market visibility.

- **Example**:
  - **Pros**:
    - Strong background in machine learning and data analytics.
    - Experience with Python and SQL.
    - Demonstrated leadership in past internships.
    - Hands-on experience with real-world data projects.
    - Bachelor's degree in AI and Data Science.
  - **Cons**:
    - Limited industry certifications.
    - Lack of project management experience.
    - No mention of soft skills or teamwork.
    - Gaps in employment history are unexplained.
    - Minimal detail on contributions to previous roles.

6.5 Working Of Open Ai In The Resume Enhancer Feature

The following Python code demonstrates how GPT is integrated into the **Resume Enhancer** feature to analyze the resume and generate the feedback discussed above:

```
def resume_GPT(resume_text):
    messages = [
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": f"""
        Analyze the following resume and provide a dictionary with 'pros' and 'cons'.
        'pros' should list 5 points that make the resume attractive.
        'cons' should list 5 areas for improvement.
        'jobs' should list 5 jobs that they can apply based on the resume.

        Resume Text:
        {resume_text}
        Response Format:
        {{
            "pros": [list of pros],
            "cons": [list of cons],
            "jobs": [list of jobs]
        }}
        """}
    ]

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=messages
    )

    return response['choices'][0]['message']['content']
```

Fig 1.3 Sample Resume Code

CHAPTER 7

7. CONCLUTION

In conclusion, Insightify addresses the challenges of modern education and career development by offering a tailored, cohesive platform that guides users through skill acquisition and career growth. By focusing on personalized tools and resources, Insightify helps individuals make informed decisions about their learning paths, identify the most relevant skills for their fields, and enhance their employability. As the demand for upskilling and continuous learning grows, Insightify empowers users to take charge of their educational and professional futures, bridging the gap between knowledge acquisition and career advancement with clarity and confidence.

REFERENCE

[1] Kumar, A., & Sharma, M. (2020). "AI-powered medical chatbots: Implications for healthcare and patient outcomes." *Journal of Medical Systems*, 44(5), 102.

[2] Bates, M., & Abbott, R. (2019). "The role of artificial intelligence in clinical decision-making: Can medical chatbots improve healthcare delivery?" *British Medical Journal (BMJ)*, 366, l4899.

[3] Bickmore, T., & Giorgino, T. (2018). "Healthcare delivery through conversational agents: A case for patient-centered medical chatbots." *Patient Education and Counseling*, 92(2), 142-148.

[4] Miotto, R., & Li, L. (2020). "AI chatbots in healthcare: Current capabilities and future possibilities." *Frontiers in Medicine*, 7, 561.

[5] Richter, F., & Flender, C. (2021). "Medical chatbots for triaging and clinical guidance: A comparative study of performance and user satisfaction." *Journal of Healthcare Informatics Research*, 5(4), 307-319.

[6] Kowatsch, T., & Nißen, M. (2017). "Improving mental health care with digital interventions: The role of chatbots in therapeutic engagement." *Journal of Medical Internet Research*, 19(6), e107.

[7] Laranjo, L., & Dunn, A. (2018). "Conversational agents in healthcare: A systematic review of medical chatbots' efficacy." *Journal of the American Medical Informatics Association (JAMIA)*, 25(9), 1248-1258.

[8] Hoermann, S., & Kramer, J. (2018). "The integration of medical chatbots into telemedicine: A path towards accessible healthcare." *Journal of Telemedicine and Telecare*, 24(5), 289-297.

[9] Kocaballi, A., & Berkovsky, S. (2020). "The rise of conversational AI in healthcare: Opportunities and challenges." *Journal of Medical Internet Research*, 22(10), e20658.

[10] Jia, J., & Feng, Y. (2021). "Evaluating the effectiveness of medical chatbots in enhancing patient education and self-management of chronic diseases." *BMC Medical Informatics and Decision Making*, 21(1), 243.