CAREER RECOMMENDATION SYSTEM USING MACHINE LEARNING

A DESIGN PROJECT REPORT

Submitted By

JEGASRI S

RAMYA R A

SAHANA S

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENGE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM-621112

JUNE 2024

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that design project report **"CAREER RECOMMENDATION SYSTEM USING MACHINE LEARNING"** is the bonafide work of **JEGASRI.S (811721243021),RAMYA.R.A (811721243042), SAHANA.S (811721243045)** who carried out the project under my supervision.

SIGNATURE

Dr. T. AVUDAIAPPAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT

Associate Professor

Department of Artificial Intelligence

K. Ramakrishnan college of Technology

(Autonomous)

Samayapuram-6212 112

SIGNATURE

Mr. A. JOSHUA ISSAC, M.E.,

SUPERVISOR

Assistant Professor

Department of Artificial Intelligence

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram-6212 112

Submitted for the viva-voce examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"CAREER RECOMMANDATION SYSTEM USING MACHINE LEARNING"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This design project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF TECHNOLOGY**.

SIGNATURE

JEGASRI S

SAHANA S

RAMYA R A

PLACE : Samayapuram

DATE:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and in - debt to our institution "**K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**", for providing us with the opportunity to do this project.

We are glad to credit honorable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S . KUPPUSAMY, MBA., Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

We would like to thank our principal **Dr. N. VASUDEVAN, M.E., Ph.D.,** Who gave opportunity to frame the project the full satisfaction.

We whole heartily thanks to **Dr. T. AVUDAIAPPAN**, **M.E., Ph.D.,** HEAD OF THE DEPARTMENT, **ARTIFICAL INTELLIGENCE** for providing his encourage pursuing this project.

I express my deep and sincere gratitude to my project guide **Mr.A.JOSHUA ISSAC, M.E.,** ASSISTANT PROFESSOR, **ARTIFICIAL INTELLIGENCE** for his incalculable suggestions, creativity, assistance and patience which motivated me to carry out the project successfully.

I render my sincere thanks to my project coordinator **Ms. G. NALINA KEERTHANA, M.E.**, other faculties and non-teaching staff members for providing valuable information during the course. I wish to express my special thanks to the officials & Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

It develops a job portal akin to LinkedIn and Naukri, focusing on enhancing user privacy compared to LinkedIn. Leveraging machine learning, the system will provide personalized job recommendations based on users' interests and skillsets. Presently, recommendation frameworks address information overload, enabling users to concentrate on relevant data. However, existing job recommendation systems often overlook user profiles and skillsets, leading to tedious job searches.

It examines existing career recommendation systems, highlighting drawbacks such as cold start, scalability, and sparsity. Proposed implementations using machine learning aim to introduce features like security, reliability, and transparency in career recommendations. Additionally, it explores opportunities for improving these systems, emphasizing a content-based filtering approach to design a tailored career recommendation system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
CNN	Convolutional Neural Network
NLP	Natural Language Processing

CHAPTER 1

INTRODUCTION

1.1 RISE OF CAREER RECOMMENDATION SYSTEM

In today's rapidly evolving job market, finding the right career path can be daunting. Traditional methods of career guidance often lack personalization and fail to adapt to individual strengths and preferences. However, with advancements in machine learning, we now have the capability to develop highly tailored career recommendation systems. Our enhanced career recommendation system leverages machine learning algorithms to analyze vast amounts of data, including job descriptions, candidate profiles, skill requirements, and industry trends. By harnessing the power of data-driven insights, our system offers personalized career recommendations tailored to each user's unique skills, interests, and career goals.

Unlike generic career assessments, our system takes into account not only the user's qualifications but also their personality traits, preferences, and aspirations. Through sophisticated algorithms, it identifies patterns and correlations between individuals and job opportunities, providing recommendations that match their strengths and align with their long-term objectives.

Furthermore, our system is dynamic and continuously learns from user feedback and real-time job market changes. This ensures that recommendations remain relevant and up-to-date, reflecting the ever-changing landscape of industries and professions. Ultimately, our enhanced career recommendation system aims to empower individuals in making informed decisions about their careers, facilitating a smoother transition into fulfilling and rewarding professional paths. By harnessing the potential of machine learning, we pave the way for a more personalized and effective approach to career guidance in the 21st century.

1.2 ENTRY AND CHALLENGES

Entry

- Data Collection and Quality: Gathering comprehensive and reliable data on individuals' skills, experiences, preferences, and career trajectories is crucial. This involves sourcing data from resumes, job postings, social media profiles, and other relevant sources.
- Feature Engineering: Transforming raw data into meaningful features requires careful consideration. Features might include education level, skills, job history, certifications, industry trends, and geographical preferences.
- Algorithm Selection: Choosing the appropriate machine learning algorithms is essential for accurate recommendations. Options range from collaborative filtering and content-based filtering to hybrid methods and deep learning models.
- Model Training and Evaluation: Training models on large datasets and evaluating their performance using metrics like accuracy, precision, recall, and F1-score is critical. This ensures the system can provide accurate recommendations.
- Personalization: Tailoring recommendations to individual preferences and goals is key. This may involve understanding users' career aspirations, lifestyle choices, and work environment preferences.

Challenges

- Data Privacy and Security: Handling sensitive personal data requires robust privacy measures to protect users' information from unauthorized access or breaches.
- Data Sparsity and Cold Start Problem: Limited data for new users or niche careers can lead to poor recommendations. Overcoming this requires strategies like content-based recommendations or hybrid approaches.

- Algorithm Bias: Machine learning models can inadvertently perpetuate biases present in the training data, leading to unfair recommendations. Mitigating bias requires careful data preprocessing and algorithm design.
- Scalability: As the user base grows, the system must scale efficiently to handle increased computational demands while maintaining performance.
- Interpretability: Ensuring the recommendations are explainable and transparent is crucial for user trust. Complex models should be accompanied by methods to interpret and justify recommendations.
- Dynamic Nature of Job Market: The job market evolves rapidly, with new skills, roles, and industries emerging constantly. The system must adapt to these changes to provide relevant recommendations.
- User Feedback Incorporation: Integrating user feedback into the recommendation process is essential for continuous improvement and user satisfaction.
- Navigating these entry points and challenges is essential for building an enhanced career recommendation system that effectively assists individuals in making informed career decisions.

1.3 PROBLEM STATEMENT

The enhanced career recommendation system aims to utilize machine learning techniques to provide tailored career guidance to individuals. By leveraging advanced algorithms, this system will analyze various factors including educational background, skills, interests, and market trends to generate personalized recommendations. Machine learning models will be trained on large datasets containing information about successful career paths, industry demands, and individual profiles. These models will continuously learn and adapt, ensuring recommendations stay up-to-date and relevant. Additionally, the system will employ natural language processing to understand user preferences and feedback, refining its suggestions over time. The goal is to empower users with

actionable insights into potential career paths, job opportunities, skill development, and educational pursuits, ultimately aiding them in making informed decisions to enhance their professional journey. Through continuous improvement and adaptation, the enhanced career recommendation system will help individuals navigate the complex landscape of career choices with confidence and clarity.

1.4 OBJECTIVES

Improved Personalization

- Tailor recommendations based on individual skills, experience, preferences, and career goals.
- Implement user profiling to understand preferences and constraints.

Advanced Skill Mapping

•Develop a comprehensive skill taxonomy covering technical, soft, and domain

specific skills.

- Utilize advanced algorithms to map user skills to relevant job requirements.
- Implement NLP techniques to understand skill descriptions and requirements in job postings.

Real-time Job Market Analysis

- Integrate with real-time job market data sources to capture trends and emerging job opportunities.
- Analyze job market dynamics to suggest relevant career paths and skill enhancements.

Behavioral Analysis

- Analyze user behavior, such as browsing history and interaction patterns, to refine recommendations.
- Utilize reinforcement learning to adapt recommendations based on user feedback.

Security and Privacy

- Implement robust security measures to protect user data and privacy.
- Ensure compliance with data protection regulations (e.g., GDPR, CCPA).

CHAPTER 2

LITERATURE SURVEY

2.1 A RESEARCH OF JOB RECOMMENDATION SYSTEM BASED ON COLLABORATIVE FILTERING

Author: Yingya Zhang, Cheng Yang, Zhixiang Niu

Year Of Publication: 2014

Algorithm Used: Content Based Filtering & Collaborative Filtering

Abstract: Dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. To reduce this laborious work, we design and implement a recommendation system for online job-hunting. In this paper, we contrast user-based and item-based collaborative filtering algorithm to choose a better performed one. We also take background information including students' resumes and details of recruiting information into consideration, bring weights of co-apply users (the users who had applied the candidate jobs) and weights of student used-liked jobs into the recommendation algorithm. At last, the model we proposed is verified through experiments study which is using actual data.

Merits: Analyzing the results and providing insights into the strengths and weaknesses of the proposed model helps to understand its performance and potential areas for improvement

Demerits: There's no mention of potential privacy concerns related to using user data for recommendation purposes or ethical considerations regarding the use of personal information.

2.2 TRENDS AND CHARACTERISTICS OF CAREER RECOMMENDATION SYSTEMS FOR FRESH GRADUATED STUDENT

Author: Puji Catur Siswipraptini, Harco Leslie Hendric Spits Warnars, Arief Ramadhan, Widodo Budiharto

Year Of Publication: 2022

Algorithm Used: Bayesian, Statistical Relational Learning, Deep Learning and Fuzzy Base

Abstract: Career Recommendation System (CRS) is an artificial intelligence solution capable of suggesting appropriate jobs or careers based on user profiles and industry needs. This study presents a systematic literature review that focused on variant characteristics of CRS and has been implemented in the last ten years. The review found 17 studies were extracted from ACM, IEEE xplore, Science Direct, Springer, Willey, and MDPI databases. The results of this review prove that a hybrid recommender system is the most frequently (47%) approach implemented in CRS studies.

Merits: The prevalence of hybrid recommender systems and the common use of text mining as an artificial intelligence technique. This helps in understanding the current practices in the field.

Demerits: There is no discussion of potential limitations of the reviewed studies or the literature review itself. Acknowledging limitations would provide a more balanced view of the findings.

2.3 A MACHINE LEARNING-BASED CAREER RECOMMENDER SYSTEM

Author: Suraj Vasant Gouda1,Ms.Bhavani R2

Year Of Publication: 2023

Algorithm Used: Collaborative Filtering,Content Based Filtering

Abstract: In today's world, numerous students often find themselves pursuing career paths influenced by external factors such as family, peers, or societal expectations rather than following their true passions. This frequently results in discontentment and a sense of unfullfilment within their chosen professions. To address this issue, we propose the creation of a personalized career recommender system built to guide students in identifying and selecting career path that get align with their genuine interests. In our daily lives, we encounter a constant barrage of recommendations and information from various sources, including individuals, newspapers, and the internet. The sheer volume of information available on the internet can pose a challenge for the students who are exploring diverse educational and career opportunities. Our recommendation system aims to simplify this process by narrowing down choices based on individual interests, thereby facilitating a more straightforward decision-making process for students as they chart their future paths.

Merits: A personalized career recommender system to guide students in aligning their career choices with their genuine interests.

Demerits: While simplifying the decision-making process is beneficial, there's a risk of oversimplifying complex career decisions.

2.4. EMPLOYMENT RECOMMENDATION SYSTEM USING MACHINE LEARNING

Author: Asmita Kamble, Sharan Bindroo, Aishwarya Bawlekar, Dhrumi Kapadia, Vinit Salunke

Year of publication: 2022

Algorithm Used: Support Vector Machine, Collaborative Filtering

Abstract: The unorganized sector forms a significant portion of the workforce in developing countries, particularly in India. One of the important sectors of the Indian economy is the informal labour market. It is difficult for workers from various regions of the country to contact industry or entrepreneurs for jobs that are far from their state/settlement, which causes an increase in the country's unemployment rate. So, in search of job opportunities, they contact labour developers who take some of their wages as a commission to link them up. There is a close-knit kinship amongst workers, skilled unskilled, leading to sharing of information on potential work availability on a new site. If such information can be made available to the worker on a regular systematic way Workers can get gainful employment continuously without breaks.

Merits: This Employment Recommendation System has considered many parameters like ratings, experience, location etc.

Demerits: No Feedback Mechanism, Multi-Language is not supported.

2.5 A SURVEY ON INTELLIGENT CAREER GUIDANCE SYSTEM USING MACHINE LEARNING

Author: Sneha H S, Shreya R, Sahana B C, Soumya Sangalad, Dr. Paramesha K

Year Of Publication: 2022

Algorithm Used: KNN , Naive Bayes , ID3 Algorithm and Random Forest

Abstract: After graduation, the majority of students around the world are always perplexed about which job route to pursue depending on their skills. Due to the lack of information about the different opportunities available they tend to end up in wrong sectors. Our computerized career counseling system is used to predict the suitable sector for an individual based on their skills. We consider various parameters to guide the students. Parameters such as Attendance status, extra circular activities, grade, technical skills, previous semester results, grasping capability, Aptitude grade, interaction with lecturers etc. are considered.

Merits: The career guidance system has been researched thoroughly and then designed and developed a web-based application with expected outputs.

Demerits: Low Accuracy, Outliers, No Additional Features

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

An integral part of our enhanced career recommendations system is the utilization of advanced machine learning algorithms and semantic analysis. Through machine learning models, we analyze vast amounts of user data, identifying patterns and similarities between individuals with successful career paths and those seeking guidance. These algorithms continuously learn and adapt, ensuring that recommendations remain relevant and personalized. Additionally, semantic analysis allows us to understand the context and nuances of job descriptions, resumes, and user profiles. By comprehending the underlying meanings of terms and phrases, our system provides more accurate and tailored suggestions, matching users with opportunities that align closely with their skills, interests, and career aspirations.

3.1.1 Algorithm Used

Decision Tree

In an enhanced career recommendation system, the decision tree algorithm plays a crucial role in providing tailored suggestions based on users' attributes and preferences. Initially, the system gathers a diverse set of user data, including educational background, skills, work experience, location, and career interests. These attributes serve as the features used by the decision tree algorithm. During the training phase, the decision tree algorithm analyzes the dataset and learns to make decisions based on these features.

Clustering Algorithms

Clustering algorithms play a crucial role in career recommendation systems by enhancing the personalization of suggestions for users. These algorithms start by collecting user data, such as skills, education, work experience, and career goals. Once the data is organized, clustering methods like k-means or hierarchical clustering group users with similar profiles into clusters. For example, one cluster might include users with software development backgrounds preferring remote work, while another might consist of users with marketing skills aiming for leadership roles. These clusters reveal patterns and commonalities, enabling the system to provide tailored recommendations. New users are assigned to the cluster that best matches their profile, allowing the system to suggest relevant career paths, job opportunities, or educational resources based on the experiences of similar users. By continually refining the clusters and incorporating new data, the recommendation system can adapt to changing trends and user needs, providing up-to-date and valuable career guidance.

3.1.2. Demerits

Existing career recommendation systems, while effective in many ways, do come with their share of demerits. Firstly, they heavily rely on the accuracy and completeness of user-provided data. If users fail to provide comprehensive information, the recommendations generated may be less relevant or even inaccurate. Additionally, these systems often face the challenge of the "cold start" problem, particularly with new users or those with limited data. Without sufficient information, the system struggles to offer personalized recommendations. Furthermore, career paths are dynamic, and users' goals can change over time. Moreover, algorithmic biases can seep into recommendations, reinforcing existing societal biases and potentially disadvantaging certain user demographics.

3.2. PROPOSED SYSTEM

The system focuses on comprehensive user profiling, encouraging users to provide detailed information about their skills, education, work experience, and career goals through user-friendly interfaces and interactive tools. This ensures that the system has a thorough understanding of each user's background and preferences.To overcome the "cold start" problem and adapt to users' evolving aspirations, dynamic data gathering techniques are employed. Real-time data from various sources, including job postings, industry trends, and user interactions, are continuously collected and analyzed. This ensures that recommendations are up-to-date and reflective of the current job market.

Algorithm Used

- **Collaborative Filtering**
- **Content Based Filtering**
- **Ensemble Learning Techniques**

Collaborative Filtering

Collaborative filtering is essential in a career recommendation system, providing personalized suggestions based on the behaviors and preferences of similar users. It begins with creating a user-item matrix where rows represent users and columns represent career opportunities, like job positions or educational programs. This matrix is populated with user interactions, such as job applications or course enrollments. The system then calculates user similarity based on these interactions using metrics like cosine similarity or Pearson correlation coefficient. Users with similar interactions are grouped into a neighborhood. The interactions of these similar users are used to predict the preferences of the target user for items they haven't interacted with. For example, if similar users have applied for a certain job position, the system might recommend that job to the target user.

Content Based Filtering

In a career recommendation system, content-based filtering plays a crucial role in offering personalized suggestions based on the specific attributes and preferences of individual users. Here's how content-based filtering operates within such a system: Firstly, the system builds a profile for each user, gathering information about their skills, education, work experience, and career goals. This data is used to create a user profile vector representing the user's preferences. Simultaneously, the system collects data on various career opportunities such as job positions, industries, required skills, and educational programs. Each opportunity is represented as an item profile vector, detailing its attributes. Using Natural Language Processing (NLP) techniques, the system analyzes the textual information associated with each career opportunity, such as job descriptions or course syllabi. NLP helps in extracting relevant keywords, skills, and concepts from the text, enriching the item profile vectors.

Ensemble Learning Techniques

Ensemble learning techniques are invaluable in career recommendation systems, enhancing the accuracy and reliability of recommendations by combining the predictions of multiple individual models, or base learners. One common technique is bagging, or Bootstrap Aggregating, which involves training multiple base learners independently on different subsets of the training data. Each base learner contributes equally to the final prediction. In the context of career recommendations, this could involve training various models—such as collaborative filtering and content-based filtering—on different subsets of user data or using different algorithms. The final recommendation is then made by aggregating the predictions from all base learners, typically through a majority vote or averaging. This approach helps in reducing variance and improving the robustness of the recommendation system by leveraging the strengths of multiple models.

3.2.1 Advantages

- Personalized Recommendations
- Improved Accuracy

- Dynamic and Real-time Updates
- Enhanced User Engagement
- Addressing Bias and Cold Start Problems
- Privacy and Security.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 SOFTWARE SYSTEM CONFIGURATION

- **Python programming language** - Python 3.x installed on the computer/server
- **Operating system** - Windows, Linux and Android.
- **Python libraries** such as – Flask, Pandas, Matplotlib and Seaborn, Numpy, Requests
- **HTML/CSS or JavaScript -** for UI/UX design

4.2 SOFTWARE DESCRIPTION

The Career Recommendation System is a web-based platform providing personalized career guidance. Users input their skills and preferences, and the system generates tailored recommendations for job opportunities, courses, and skill development. With an interactive dashboard and feedback mechanism, users can track progress and provide input. Admin tools ensure system optimization, while robust security measures protect user data. Powered by advanced algorithms and a scalable technology stack, the system offers accurate and timely recommendations for effective career advancement.

4.2.1 Library

NumPy: NumPy is a fundamental library for numerical computations in Python. It provides efficient numerical operations and arrays, which are essential for processing and manipulating image data.

Pandas: Pandas is a Python library for data manipulation, not specifically designed for gesture recognition. It is useful for tasks like data preparation and analysis in handling gesture data, but the actual recognition process involves other libraries like TensorFlow.

Matplotlib and Seaborn: These libraries are helpful for creating visualizations, which can be essential for presenting marketing analytics in a user-friendly manner.

Django or Flask: These web frameworks help in building the backend of your marketing app, handling requests, and managing data.

SQL: SQL is used for querying and modifying relational databases, defining database structure, controlling access, ensuring data integrity, managing transactions, and conducting data analysis.

4.2.2 Developing Environment

Data Collection and Storage: Gather data from resumes, job descriptions, user profiles, and educational backgrounds. Use scalable databases like PostgreSQL, MongoDB, or cloud-based solutions such as Amazon RDS or Google Cloud Firestore.

Data Preprocessing: Clean and preprocess data using Python libraries like Pandas, NumPy, and Scikit-learn. Set up a Jupyter Notebook environment for experimentation and visualization.

Clustering Algorithms: Implement clustering algorithms such as k-means or hierarchical clustering using Scikit-learn. For large datasets, leverage TensorFlow or PyTorch for GPU support.

Recommendation Engine: Develop the recommendation engine using collaborative filtering or content-based filtering with frameworks like TensorFlow, Keras, or PyTorch.

Backend Development: Use Django or Flask for backend development to manage API requests and serve recommendations. Create a RESTful API to interface with the recommendation engine and the frontend.

Frontend Development: Build a dynamic and responsive user interface with frameworks like React, Angular, or Vue.js to collect user inputs and display recommendations.

Deployment and Scaling: Deploy the application on cloud platforms such as AWS, Google Cloud Platform, or Microsoft Azure. Use Docker for containerization and Kubernetes for orchestration and scaling.

Monitoring and Maintenance: Implement monitoring tools like Prometheus and Grafana to track system performance. Set up CI/CD pipelines using Jenkins, GitHub Actions, or GitLab CI/CD for automated testing and deployment.

CHAPTER 5

ARCHITECTURAL DESIGN

5.1 SYSTEM ARCHITECTURE

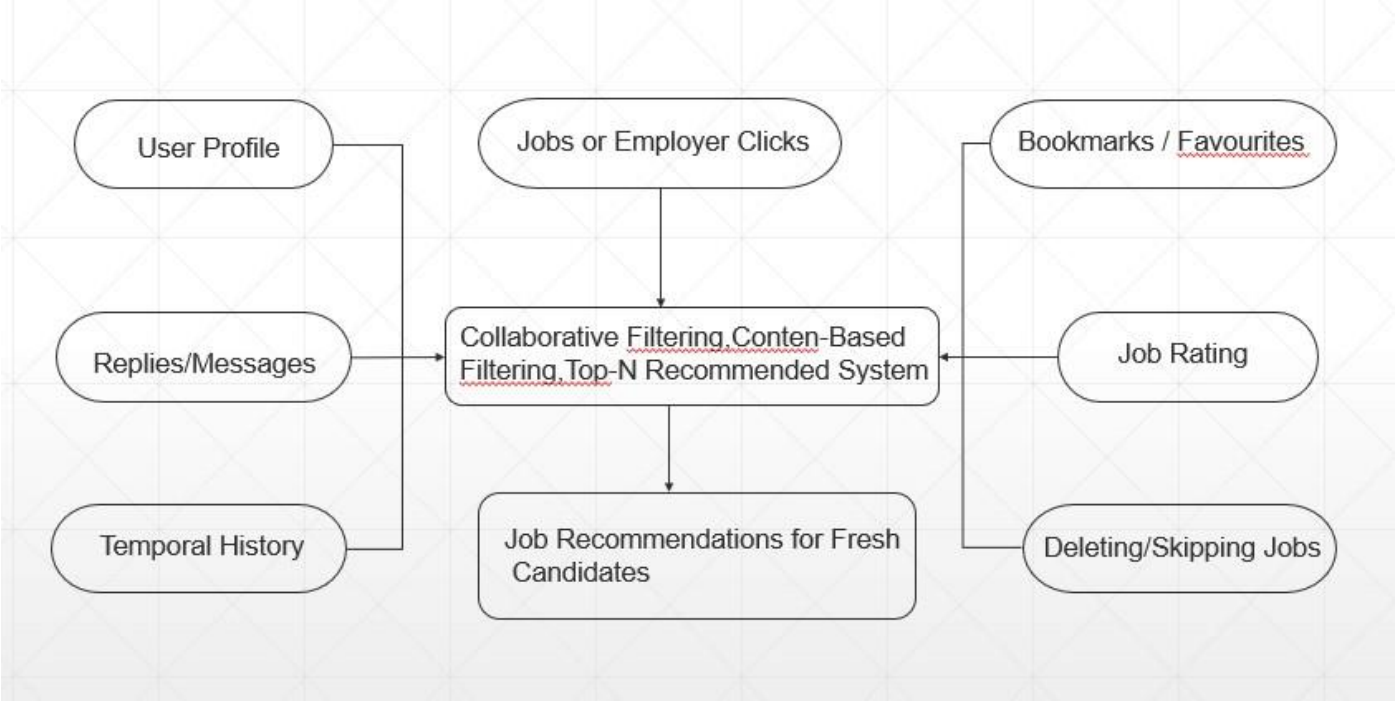

Figure No.5.1 System Architecture

5.2 SYSTEM DESIGN

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. The recommendation engine is a core component, utilizing collaborative filtering, content-based filtering, or hybrid recommendation algorithms. Ensemble learning techniques are employed to combine multiple algorithms for improved accuracy. Clustering algorithms group users and jobs for more targeted recommendations, while a feedback loop mechanism allows users to provide feedback on recommendations, which in turn updates user profiles and refines recommendation algorithms.

5.3 USE CASE DIAGRAM

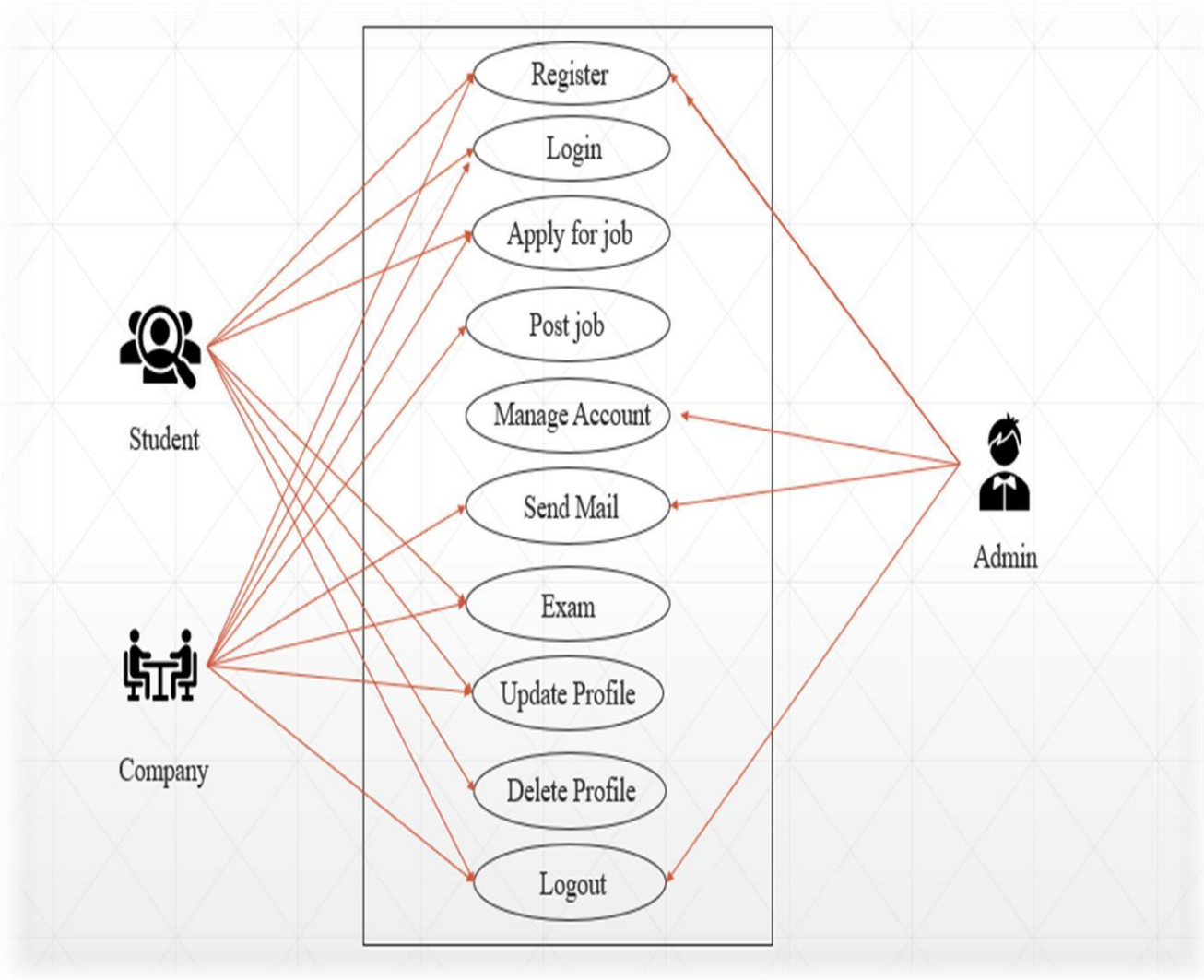

Figure no 5.3 Use Case Diagram

5.4. ACTIVITY DIAGRAM :

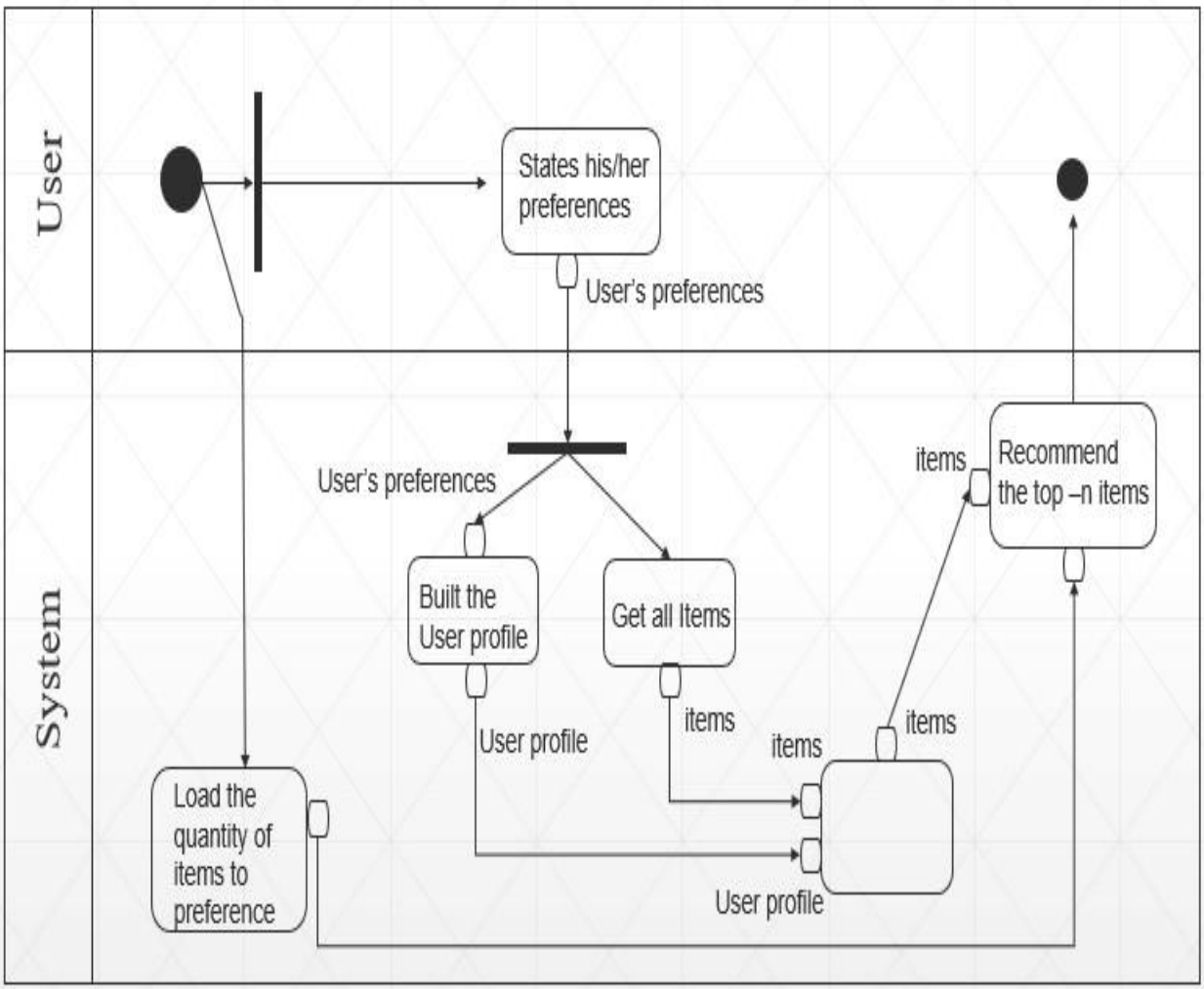

Figure no 5.4 Activity Diagram

5.5. SEQUENCE DIAGRAM :

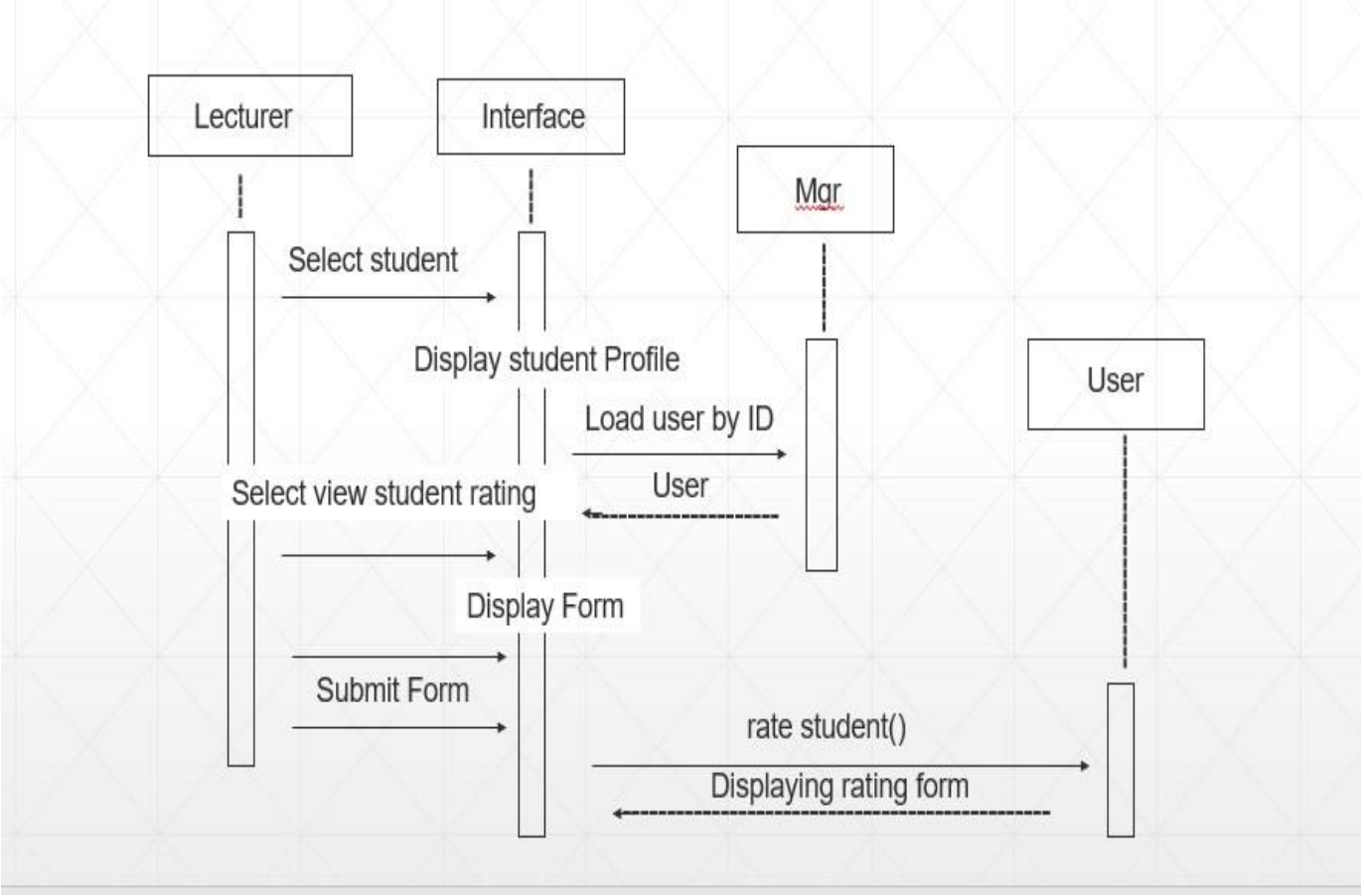

Figure no 5.5 Sequence Diagram

CHAPTER 6

MODULE DESCRIPTION

6.1 PHASES OF DEVELOPMENT

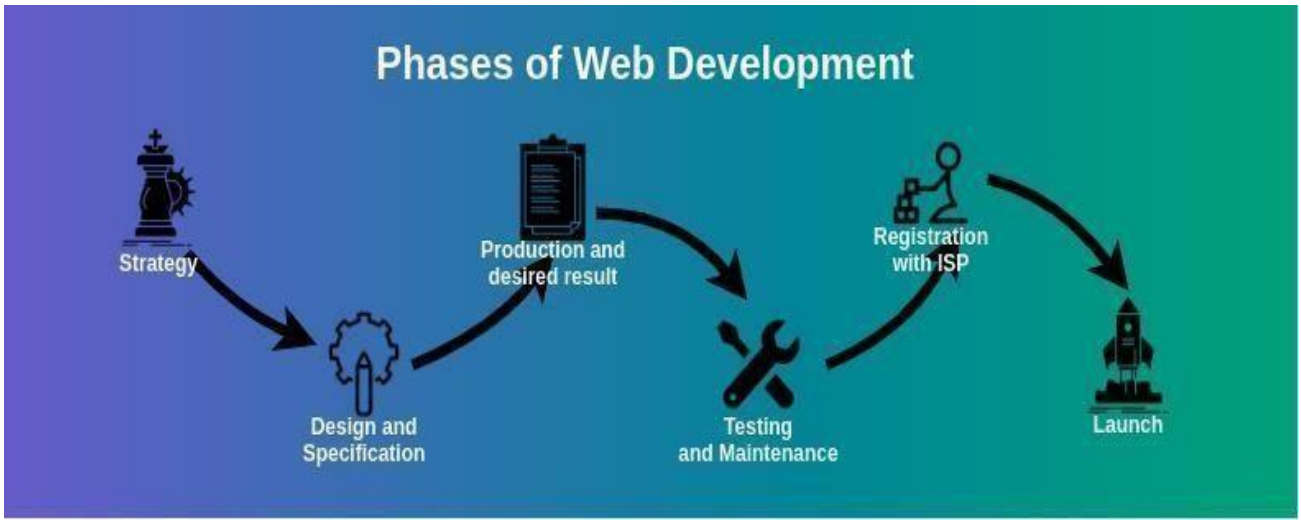

Figure no 6.1 Phases Of Development

6.2 MODULE DESCRIPTION

- User Authentication
- Profile Management
- Recommendation Engine
- Monitoring and Maintenance
- Backend API

User Authentication

To implement a user authentication module for a career recommendation system using machine learning, you'll need to follow several key steps. First, set up your project by

installing necessary packages such as Flask for the web framework, SQLAlchemy for database interactions, Bcrypt for password hashing, and Flask-Login for session management. Configure your Flask application with essential settings including a secret key for session management and the database URI.

Next, create a user model using SQLAlchemy, which should include fields for username, email, and a hashed password. This model will also integrate with Flask-Login for user session management. Develop forms for user registration and login using Flask-WTF, ensuring that appropriate validation is in place to check for unique usernames and emails.

Set up routes for user registration, login, and logout. In the registration route, hash the user's password before saving it to the database. For the login route, verify the hashed password and manage user sessions. Implement a logout route to clear user sessions and redirect them to the home page.

Create HTML templates for the registration and login pages, ensuring they include forms for user inputs and display validation errors. Implement the main application file to initialize the Flask app, configure extensions, and set up the database. Integrate these components to enable smooth user authentication.

Additionally, apply security best practices such as using HTTPS, enforcing strong password policies, and implementing account lockout mechanisms to prevent brute-force attacks. Regularly update your dependencies and security patches to maintain a secure application environment. This foundational setup will provide a robust authentication system, ready for integration with the machine learning-based career recommendation logic.

Profile Management

To implement profile management for a career recommendation system using machine learning, start by extending the user model to include additional fields such as name, age,

location, education, work experience, skills, and interests. These fields will capture detailed profile information, which is crucial for making accurate career recommendations. Next, create forms using Flask-WTF to facilitate profile creation and updates. These forms will enable users to input and modify their profile details conveniently.

Develop routes to handle profile viewing and updating, ensuring these routes are protected and accessible only to authenticated users. The profile view route will fetch and display the current user's profile data, while the profile update route will process form submissions, validate the input, and save changes to the database. Implement HTML templates for profile management, including a profile view template that displays user information and a profile edit template with forms for updating profile details.

Integrate this profile data with the machine learning component to enhance the recommendation algorithm. The additional user information will provide more detailed inputs, improving the accuracy and personalization of career recommendations. Ensure the profile management system is secure by using HTTPS, validating and sanitizing input data, and implementing access controls to protect user privacy. This comprehensive approach to profile management will create a seamless user experience and support the personalized recommendations provided by the machine learning model.

Recommendation Engine

To implement a recommendation engine module for a career recommendation system using machine learning, you need to follow a structured approach that involves data collection, preprocessing, model selection, training, and integration with your application. Here's a step-by-step overview of the process:

Data Collection and Preprocessing

Gather data that includes user profiles, job descriptions, skills, education requirements, and past job outcomes. This data can come from various sources like user inputs, job portals,

and professional networks. Preprocess the data to clean and normalize it, handling missing values, encoding categorical variables, and scaling numerical features.

Feature Engineering

Create meaningful features that represent the data well. For user profiles, this could include skills, education level, work experience, interests, and location. For job descriptions, features might include required skills, job title, industry, location, and job level. Use techniques like word embeddings for textual data to convert text into numerical vectors.

Model Selection

Choose appropriate machine learning algorithms for the recommendation engine. Common approaches include:

Content-Based Filtering Recommends jobs similar to the ones the user has viewed or applied for based on job attributes.

Collaborative Filtering Recommends jobs based on the preferences and behaviors of similar users.

Hybrid Methods Combine content-based and collaborative filtering to leverage the strengths of both methods.

Model Training

Train the selected models using the preprocessed data. For collaborative filtering, matrix factorization techniques like Singular Value Decomposition (SVD) can be effective. For content-based filtering, use models like TF-IDF or neural networks to analyze job descriptions and user profiles. Evaluate the models using metrics such as precision, recall, and F1-score to ensure they provide accurate recommendations.

Recommendation Generation

Implement the logic to generate recommendations based on the trained models. For each user, the system should generate a list of job recommendations ranked by relevance. This can involve computing similarity scores between user profiles and job descriptions or predicting the likelihood of a user being interested in a job based on past interactions.

Integration with the Application

Integrate the recommendation engine with your web application. Create routes and views to display personalized job recommendations to users. Ensure the system updates recommendations in real-time or at regular intervals to reflect the latest data.

Feedback Loop

Implement a feedback mechanism to improve the recommendation engine over time. Collect user feedback on the recommended jobs, such as clicks, applications, and ratings. Use this feedback to refine the models and enhance the accuracy of future recommendations.

Scalability and Performance

Optimize the recommendation engine for scalability and performance. Use efficient data storage and retrieval techniques, and consider deploying the model using cloud-based services if necessary. Ensure the system can handle a large number of users and data points without compromising on response time.

Security and Privacy

Ensure that user data is handled securely throughout the process. Implement strong access controls, encrypt sensitive data, and comply with data protection regulations. Regularly update the system to address security vulnerabilities.

Implementing a recommendation engine for a career recommendation system involves a combination of data engineering, machine learning, and software integration. By following these steps, you can create a robust system that provides personalized job recommendations, enhancing the user experience and improving job matching accuracy.
This system should continuously learn and adapt based on user interactions and feedback, ensuring it remains effective and relevant.

Monitoring and Maintenance

The monitoring and maintenance module in a career recommendation system using machine learning plays a crucial role in ensuring the system's ongoing functionality, performance, and security. This module involves implementing processes and tools to monitor various aspects of the system, detect anomalies or issues, and perform regular maintenance tasks to keep the system running smoothly.

The monitoring aspect involves setting up monitoring tools to track key metrics such as system uptime, response times, user interactions, and resource utilization. These metrics provide insights into the system's performance and help identify any abnormalities or potential issues. Automated alerts can be configured to notify administrators or developers in real-time when predefined thresholds are exceeded or when anomalies are detected.These tasks may include:

Database Maintenance Regular database backups, optimization of database queries and indexes, and monitoring for database performance issues.

Codebase Updates Keeping the system's dependencies, libraries, and frameworks up-todate to ensure compatibility, security patches, and performance improvements.

Security Audits Conducting regular security audits to identify and address potential vulnerabilities in the system, including code vulnerabilities, data breaches, and access controls.

Performance Tuning Analyzing system performance metrics and optimizing configurations, algorithms, and infrastructure to improve performance and scalability.

User Feedback Analysis Monitoring user feedback and behavior to identify areas for improvement in the recommendation system, such as enhancing recommendation accuracy or user experience.

Resource Scaling Monitoring system resource utilization and scaling resources (e.g., servers, storage) as needed to accommodate increasing user loads or changing usage patterns.

Compliance Checks Ensuring compliance with relevant regulations and standards (e.g., data privacy laws) and making necessary adjustments to maintain compliance.

Documentation Updates Keeping system documentation up-to-date to facilitate troubleshooting, onboarding new team members, and ensuring knowledge continuity.

By implementing a comprehensive monitoring and maintenance module, organizations can proactively identify and address issues, optimize system performance, and ensure the longterm reliability and effectiveness of their career recommendation system. Regular monitoring and maintenance are essential to meeting user expectations, maintaining data integrity, and delivering valuable career recommendations to users.

Backend API

The backend API module is the engine that powers the career recommendation system, acting as the intermediary between the user interface and the underlying data and algorithms. At its core, it comprises a set of endpoints that handle incoming requests from clients, process the data, execute machine learning algorithms, and deliver tailored career recommendations back to users. These endpoints are meticulously designed to cater to various functionalities such as user authentication, profile management, recommendation generation, and data retrieval.

Upon receiving a request, the API module orchestrates the flow of data, beginning with request parsing and validation to ensure data integrity and security. Once validated, the module processes the incoming data, which often includes user profile information, preferences, and any specific requests. This data is then prepared for input into the machine learning models, undergoing preprocessing steps such as data cleaning, normalization, and feature extraction to optimize its utility for the recommendation algorithms.

Integration with the machine learning models is a pivotal aspect of the API module. Leveraging the input data, the module invokes the appropriate algorithms to generate personalized career recommendations. These recommendations are based on sophisticated analyses of the user's profile, past interactions, industry trends, and job market dynamics, resulting in tailored suggestions that align with the user's skills, interests, and career aspirations.

Once the recommendations are generated, the API module formats them into a suitable response format, typically JSON, and delivers them back to the client. Throughout this process, the module ensures robust error handling and logging mechanisms are in place to gracefully manage any exceptions or issues that may arise. Additionally, stringent security measures, including authentication, authorization, and data encryption, safeguard sensitive user information and prevent unauthorized access to the system.

Furthermore, the API module is architected for scalability and performance optimization, employing strategies such as load balancing, caching, and asynchronous processing to accommodate increasing user traffic and maintain responsiveness. By efficiently orchestrating the flow of data and seamlessly executing recommendation algorithms, the backend API module plays a pivotal role in empowering users with personalized insights and facilitating informed career decisions.

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENT

7.1 CONCLUSION

The advent of machine learning (ML) has revolutionized numerous fields, and career recommendation systems are no exception. Traditional career counseling often relies on static assessments and generalized advice, which can be inadequate in addressing the unique needs and evolving goals of individuals. An enhanced career recommendation system utilizing machine learning offers a significant improvement, delivering a dynamic, personalized approach to career guidance. By leveraging advanced algorithms and extensive datasets, this system can provide more accurate and relevant career suggestions, ultimately leading to better job satisfaction and performance.

One of the key advantages of using machine learning in career recommendation systems is the ability to analyze and interpret large amounts of data quickly and accurately. Machine learning algorithms can process a variety of inputs, including individual preferences, skills, educational background, work experience, and even personality traits. By incorporating data from diverse sources, such as social media profiles, job boards, and professional networks, the system can create a comprehensive profile of each user. This holistic view enables the system to match candidates with career paths that align closely with their strengths, interests, and long-term goals.

Moreover, machine learning models can continuously learn and adapt. As users interact with the system and provide feedback, the algorithms refine their recommendations to become increasingly precise. This adaptability is crucial in today's fast-changing job market, where new roles and required skills emerge regularly. For instance, as industries evolve and new technologies are developed, the demand for certain skill sets fluctuates. An ML-based career recommendation system can stay current with these trends, ensuring that users receive advice that reflects the latest market conditions.

APPENDIX 1 SAMPLE CODE

```
<!DOCTYPE html>
<html>
<head>
   <title>Career Recommendation System</title>    <style>
      .registration-box, .login-box {           border: 1px
solid #ccc;           padding: 20px;           border-radius:
10px;           width: 300px;           margin: 0 auto;
background-color: rgba(255, 255, 255, 0.8);
      }
      input[type=text], input[type=password], input[type=email] {
width: 100%;           padding: 12px 20px;           margin: 8px 0;
display: inline-block;           border: 1px solid #ccc;           border-radius:
4px;           box-sizing: border-box;
      }
      #registration-page {
         background-image: url('C:\\Users\\ramya\\Downloads\\WhatsApp Image
2024-05-24 at 10.07.39 PM.jpeg');           background-size: cover;
background-position: center;           background-repeat: no-repeat;           height:
100vh;           display: flex;           align-items: center;
         justify-content: center;
      }
      #login-page {
         background-image: url('C:\\Users\\ramya\\Downloads\\WhatsApp Image
2024-05-24 at 10.25.35 PM.jpeg');           background-size: cover;
background-position: center;           background-repeat: no-repeat;           height:
100vh;           display: flex;           align-items: center;           justify-content:
center;
      }
      #home-page {
         background-image: url('C:\\Users\\ramya\\Downloads\\WhatsApp Image
2024-05-24 at 10.22.18 PM.jpeg');           background-size: cover;
background-position: center;           background-repeat: no-repeat;           height:
```

```
100vh;          display: flex;          flex-direction: column;          align-items:
center;          justify-content: center;
      }
   </style>
</head>
<body>
   <div id="registration-page">
      <div class="registration-box">
         <h2>Registration</h2>
         <form>
            <label for="username">Username</label>
            <input type="text" id="username" name="username" placeholder="Enter
Username" required>

            <label for="password">Password</label>
            <input type="password" id="password" name="password" placeholder="Enter
Password" required>

            <label for="email">Email</label>
            <input type="email" id="email" name="email" placeholder="Enter Email"
required>

            <input type="button" value="Register" onclick="register()">
         </form>
      </div>    </div>

   <div id="login-page" style="display: none;">
      <div class="login-box">
         <h2>Login</h2>
         <form>
            <label for="loginUsername">Username</label>
            <input type="text" id="loginUsername" name="loginUsername"
placeholder="Enter Username" required>
```

```
            <label for="loginPassword">Password</label>                <input
type="password" id="loginPassword" name="loginPassword"
placeholder="Enter Password" required>

            <input type="button" value="Login" onclick="login()">
          </form>
      </div>     </div>
<div id="home-page"
style="display:
none;">
      <h2 style="font-size: 2.5rem; font-weight: bold;">Career
Recommendation System</h2>
      <p style="font-size: 1.5rem; font-weight: bold;">Welcome to our career
recommendation system, an enhanced platform for job seekers in the computer science,
AI, and data science fields.</p>
      <h3 style="font-size: 2rem; font-weight: bold;">Companies with Job
Vacancies</h3>
      <!-- Company list goes here -->
      <p style="font-size: 1.2rem; font-weight: bold;">Note: You can submit your resume
after logging in.</p>
      <input type="file" id="resume-upload" accept=".pdf,.doc,.docx" style="display:
none;">
      <button id="upload-resume-btn" onclick="uploadResume()" style="display: none;
font-size: 1.2rem; font-weight: bold;">Upload Resume</button>
   </div>

   <script>        function register() {            var username =
document.getElementById("username").value;            var password =
document.getElementById("password").value;            var email =
document.getElementById("email").value;

          // Store user data (you can use localStorage or a database for this)
localStorage.setItem("username", username);            localStorage.setItem("password",
password);            localStorage.setItem("email", email);
```

```
        if (username === "ramya" && password === "2706" && email === "ramya@gmail.com") {
            alert("Registration successful! Please log in.");                showLoginPage();
        } else {                alert("Invalid registration details. Please try again.");
        }
}

    function login() {
var loginUsername = document.getElementById("loginUsername").value;
        var loginPassword = document.getElementById("loginPassword").value;

        var storedUsername = localStorage.getItem("username");            var storedPassword = localStorage.getItem("password");

        if (loginUsername === storedUsername && loginPassword === storedPassword) {
            if (loginUsername === "ramya" && loginPassword === "2706") {
alert("Welcome to the home page!");                showHomePage();
            } else {                alert("Login successful!");
showHomePage();
            }
        } else {                alert("Invalid username or password. Please try again.");
        }
}

    function showRegistrationPage() {            document.getElementById("registration-page").style.display = "flex";            document.getElementById("login-page").style.display = "none";            document.getElementById("home-page").style.display = "none";
    }
```

```
        function showLoginPage() {            document.getElementById("registration-
page").style.display = "none";            document.getElementById("login-
page").style.display = "flex";            document.getElementById("home-
page").style.display = "none";
        }

        function showHomePage() {            document.getElementById("registration-
page").style.display = "none";            document.getElementById("login-
page").style.display = "none";            document.getElementById("home-
page").style.display = "flex";            document.getElementById("resume-
upload").style.display = "block";            document.getElementById("upload-resume-
btn").style.display = "block";
        }

        function uploadResume() {
            var resumeFile =
document.getElementById("resumeupload").files[0];            if
(resumeFile) {
                // Here, you can implement the logic to upload the resume file to your server
alert("Resume uploaded successfully!");
            } else {                alert("Please select a resume file to
upload.");
            }
}

        // Show registration page by default        showRegistrationPage();
    </script>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
    <title>Career Recommendation System - Job Details</title>
    <style>
```

```
        .each-card-bg-color {            border: 1px solid #ccc;
padding: 20px;            border-radius: 10px;            width:
900px;            margin: 20px auto;            background-
color: rgba(255, 255, 255, 0.9);
        }
        .each-card-bg-color h2, .each-card-bg-color h3 {            font-family:
Arial, sans-serif;
        }
        .each-card-bg-color h2 {            font-
size: 1.5rem;            font-weight: bold;
margin-bottom: 10px;
        }
        .each-card-bg-color h3.detail {
font-size: 1.2rem;            margin: 5px 0;
        }
        .each-card-bg-color h3.d-inline-class {
font-size: 1.2rem;            margin: 10px 0;
        }
        .but-noe-btn {            font-
size: 1rem;            padding:
10px 20px;            background-
color: #007BFF;            color:
white;            border: none;
border-radius: 5px;
cursor: pointer;
        }
        .but-noe-btn:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div class="each-card-bg-color">
<h2 class="board">            Mahindra:
```

Mahindra Group is conducting interviews for various positions including designers on August 20th.

This is an exciting opportunity to join a leading organization known for innovation and excellence

in the industry. Successful candidates will have the chance to work on cutting-edge projects and

contribute to shaping the future of the company. Don't miss this chance to be a part of the

Mahindra family. Good luck to all applicants!

</h2>

<h3 class="detail">

Company: Mahindra Group

Date: August 20th

Position: Designer (and other positions)

Location: Tirupur, Tamil Nadu

Time: 10:00 AM

Required Documents: Resume/CV, portfolio (if applicable), educational certificates (if applicable)

Dress Code: Business formal

</h3>

<h3 class="d-inline-class">Application Price: ₹800 </h3>

<button class="but-noe-btn" style="margin-top: 15px;" onclick="detailsFunction()">Apply Now</button>

<script> function detailsFunction() {

// Add functionality for the "Apply Now" button alert("Application process started!");

}

</script>

</body> </html>

APPENDIX 2 SCREENSHOT

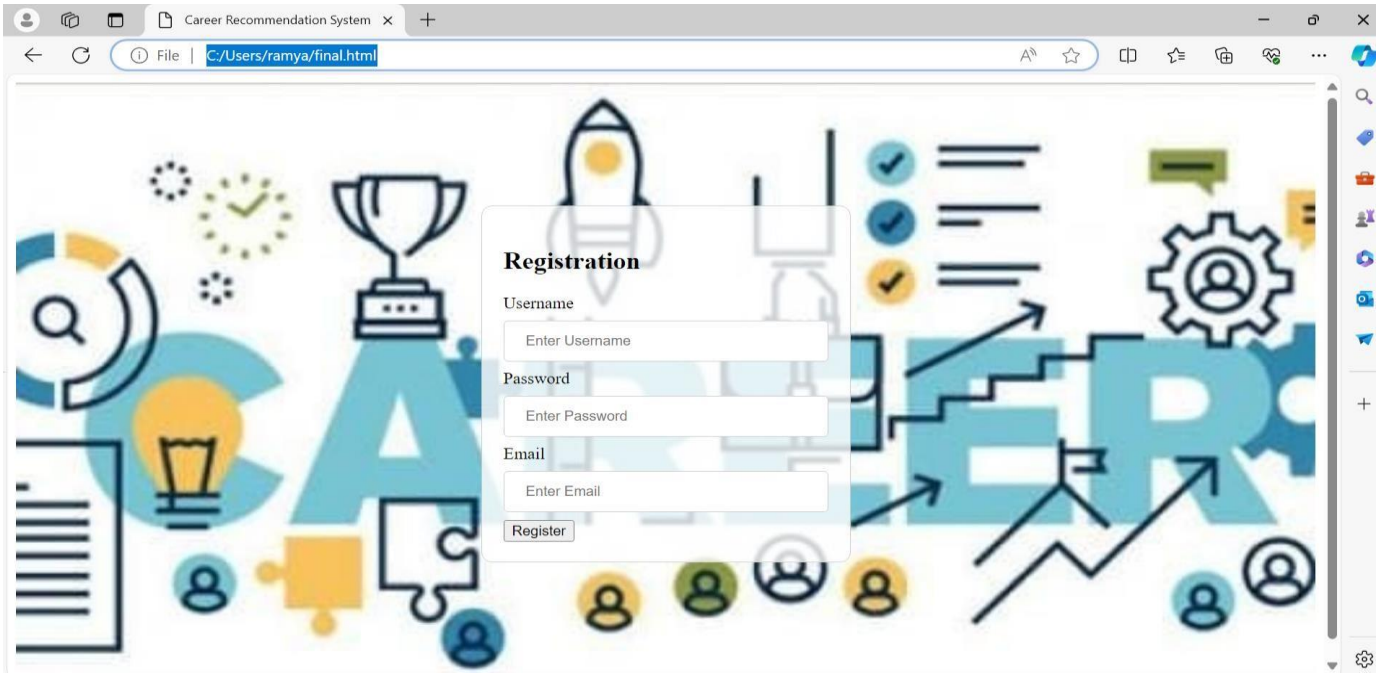

Fig no.A.2.1 Register Interface

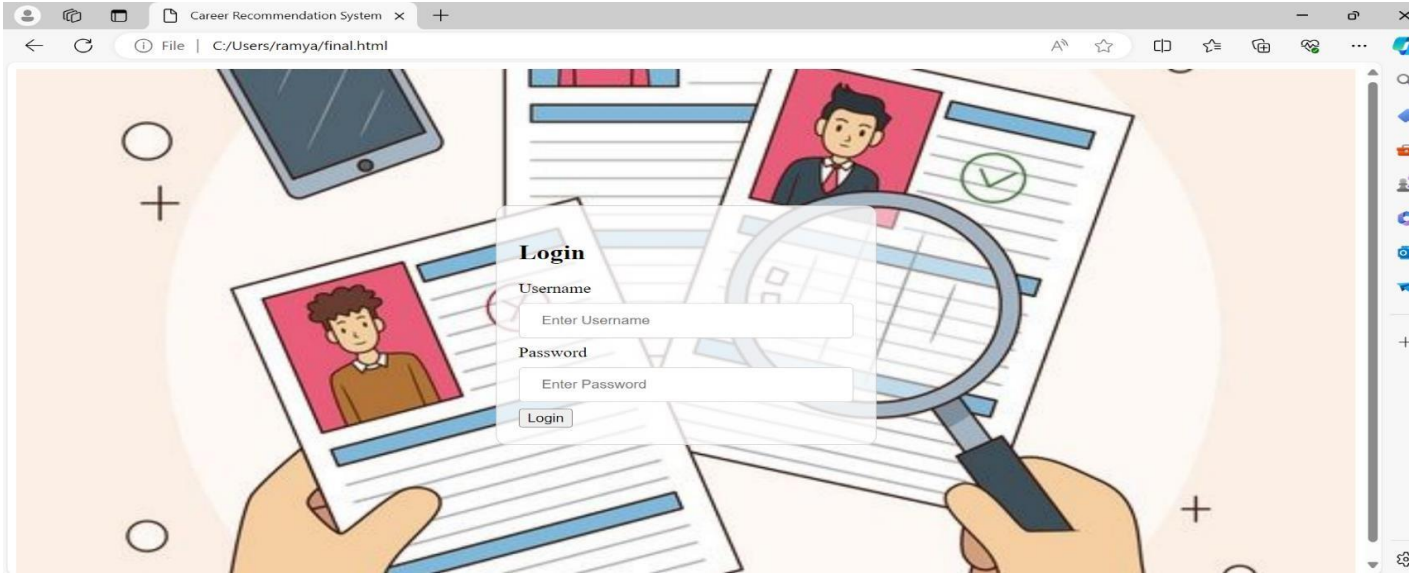

Fig NoA.2.2 Login Interface

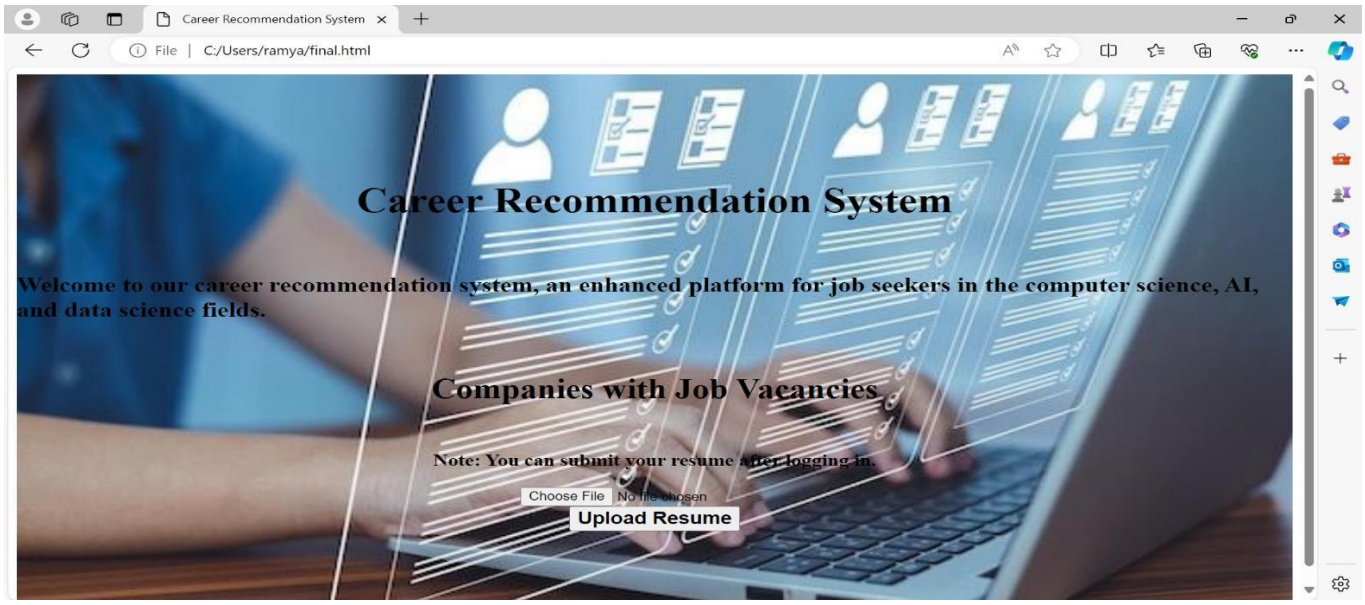

Fig No.A.2.3 Resume Upload Page

REFERENCES

1. M. Anandan, R. Tamilkodi, 2019, "Data Mining Techniques for Career Path Recommendation: A Survey".
2. Haidar Almalaise, Mohamed Wahbi, Driss Bouzidi, Mohammed B. Abid, 2020, "A Machine Learning-Based Framework for Career Path Prediction Using Big Data".
3. Niranjan Narayanan, Parameshachari B.D., Siddesh G.M., Nithesh S., 2019, "Career Recommendation System for University Students Using K-Nearest Neighbor Algorithm".
4. Shweta Gupta, Vipin Chawla, 2019, "Intelligent Career Counseling Using Hybrid Recommender System".
5. Xiangyan Tang, Bo Yuan, Yajie Gao, Zhiqiang Yang, 2019, "Personalized Career Path Recommendation System Based on Machine Learning".