CAREER DENDROGRAM

SUBMITTED IN PARTIAL FULFILLMENT FOR THE
REQUIREMENT OF THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE

Submitted by

SHAYAM (2100290120154)
SHUBHAM VERMA (2100290120161)
RAJEEV RANJAN (2100290120136)
RISHABH YADAV (2100290120143)

Supervised by
Mr. PROF. PAWAN KUMAR PAL
Assistant Professor

Session 2024-25

DEPARTMENT OF COMPUTER SCIENCE
KIET GROUP OF INSTITUTIONS, GHAZIABAD
(Affiliated to Dr. A. P. J. Abdul Kalam Technical University,

Lucknow, U.P., India)

May 2025

DECLARATION

We hereby declare that this submission is our work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Shayam
Roll No.:2100290120154

Signature

Name: Shubham Verma
Roll No.:2100290120161

Signature

Name: Rajeev Ranjan
Roll No.:2100290120136

Signature

Name: Rishabh Yadav
Roll No.:2100290120143

Date:

CERTIFICATE

This is to certify that Project Report entitled “Career Dendrogram” which is submitted by Rajeev Ranjan, Rishabh Yadav, Shayam, and Shubham Verma submitted in partial achievement of the requirement for the award of the diploma B.Tech. Inside the department of laptop technology of Dr. A.P.J. Abdul Kalam Technical College, Lucknow, is a file of the candidates' paintings finished by them under my supervision. The matter embodied on this record is original and has not been submitted for the award of some other degree.

Date:

Supervisor
Mr. Pawan Kumar Pal
Assistant Professor
CS Department

ACKNOWLEDGEMENT

It gives me great pleasure to present the work undertaken in the course of B.Tech. Fourth year. I owe a special debt of gratitude to **Mr. Pawan Kumar Pal**, Assistant Professor, Computer Science, KIET Group of Institutions, Delhi- NCR, Ghaziabad, for his steady guidance and steering during my studies. His sincerity, thoroughness, and perseverance have been a continuous supply of inspiration for me. It's miles best her cognizant efforts that my endeavors have visible the light of day.

I additionally take the opportunity to renowned the contribution of **Dr. Ajay Kumar Srivastava**, Head of the Department of Computer Technology, KIET group of institutions, Delhi-NCR, Ghaziabad, for his complete assistance and help during the improvement of the mission. I also do not like to overlook the possibility of acknowledging the contribution of all of the branch's school participants for their kind assistance and cooperation at some point in the improvement of our undertaking.

Remaining, however, now not least, I acknowledge Dr. Manoj Goel, Joint Director, KIET organization of institutions, Delhi-NCR, Ghaziabad, for his financial and moral guidance for the finishing touch of this venture.

Date:

Signature:

Name: Shayam
Roll No.: 2100190120154

Name: Shubham Verma
Roll No.: 2100290120161

Name: Rajeev Ranjan
Roll No.: 2100290120136

Name: Rishabh Yadav
Roll No.: 2100290120143

ABSTRACT

Career Dendrograms is a machine learning-based system designed to analyse and visualize professional growth trajectories, offering a centralized platform for individuals, educational institutions, and organizations. The platform utilizes a random forest algorithm to provide personalized career predictions based on users' skills, education, and work experience. It aggregates data from various sources to uncover patterns, highlight skill gaps, and suggest potential career paths through hierarchical dendrogram visualizations.

This innovative tool assists individuals in making informed career decisions, enables educational institutions to align their curricula with market demands, and supports organizations in refining hiring and training strategies. By leveraging cutting-edge technologies and data-driven insights, **Career Dendrograms** promotes strategic career planning and workforce development, addressing the dynamic needs of today’s job market.

TABLE OF CONTENTS

Page No.

DECLARATION		ii
CERTIFICATE		iii
ACKNOWLEDGEMENT		iv
ABSTRACT		v
LIST OF ABBREVIATIONS		viii
SDG MAPPING WITH JUSTIFICATION		ix
DIAGRAM FLOW TABLE		x
LIST OF TABLES		xi
CHAPTER 1 INTRODUCTION		1- 3
1.1	Introduction to Project	1
1.2	Project Category	2
1.3	Objectives	2
CHAPTER 2 LITERATURE REVIEW		4-6
2.1	Literature Review	4
2.2	Research Gaps	5
2.3	Problem Formulation	5
CHAPTER 3 PROPOSED SYSTEM		7-8
3.1	Proposed System	7
3.2	Unique Features of The System	8
CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION		9-24
4.1	Feasibility Study	9
4.2	Software Requirement Specification	10
4.3	SDLC Model Used	14
4.4	System Design	15
4.5	Database Design	22

CHAPTER 5 IMPLEMENTATION 25-33
5.1 Introduction Tools and Technologies Used. 25

CHAPTER 6 TESTING, AND MAINTENANCE 34-40
6.1 Testing Techniques and Test Cases Used 34
6.2 Test Case 37

CHAPTER 7 RESULTS AND DISCUSSIONS 41-51
7.1 Description of Modules with Snapshots 41
7.2 Key Findings of the Project 48
7.3 Brief Description of Database with Screenshots 49

CHAPTER 8 CONCLUSION AND FUTURE SCOPE 52-53

REFERENCES
PLAGIARISM REPORT

LIST OF ABBREVIATIONS

The following abbreviations are used throughout the **Career Dendrogram** report:

- **API** – Application Programming Interface
- **QR** – Quick Response
- **UI** – User Interface
- **UX** – User Experience
- **DBMS** – Database Management System
- **NO SQL** – Not Only Structured Query Language
- **JWT** – JSON Web Token
- **CI/CD** – Continuous Integration and Continuous Deployment
- **OTP** – One-Time Password
- **HTTP** – Hyper Text Transfer Protocol
- **HTTPS** – Hyper Text Transfer Protocol Secure
- **SDLC** – Software Development Life Cycle
- **RAM** – Random Access Memory
- **CPU** – Central Processing Unit
- **SSL** – Secure Sockets Layer
- **IDE** – Integrated Development Environment
- **CSS** – Cascading Style Sheets
- **HTML** – Hyper Text Markup Language

SDG MAPPING WITH JUSTIFICATION

SDG 4: Quality Education: Helps individuals access structured career paths, fostering lifelong learning and skill development.

SDG 5: Gender Equality: Ensure unbiased career recommendations , promoting equal opportunities for all genders.

SDG 8: Decent Work & Economic Growth**:** Guides individuals toward sustainable, future ready careers, supporting economic growth.

SDG 9: Industry, Innovation, Technology and Infrastructure: This career recommendation system for students is a industry innovation, as it provided an online platform by which people can predict their career online all over India at any time without any issue. We can further extend this innovation to integrate AI and ML in it to provide suggestions to the users.

SDG 10: Reduce Inequality: Provides equitable career guidance, reduce disparities across economic backgrounds.

SDG 17: Partnerships for the Goals: The project requires collaboration between governments and tech sectors. Can integrate collaborations with educational institutions and employers to enhance career success.

DIAGRAM FLOW TABLE

Diagram / Figure Name	Page No.
Fig: 4.3 – Agile SDLC	15
Fig: 4.4 – System Design	17
Fig: 4.4.1(a) – Level-0 DFD	18
Fig: 4.4.1(b) – Level-1 DFD	19
Fig: 4.4.1(c) – Level-2 DFD	20
Fig: 4.4.2(a) – Use Case Diagram	21
Fig: 4.5(a) – Database Design (under Database)	24
Fig: 5.1 – Overview of Tech used	31
Fig: 7.1 – Landing Page	43
Fig: 7.2 – About Us Page	44
Fig: 7.3 – Signup Page	44
Fig: 7.4 – Login Page	44
Fig: 7.5 – Recommendation Page	45
Fig: 7.6 – Output	45
Fig: 7.7 – Home Page	46
Fig: 7.8 – API call using Thunder client	47
Fig: 7.9 – User Database Snapshots	51

LIST OF TABLES

Table. No	Description	Page No.
2.1	Literature Review	4
6.1	User Registration	37
6.2	Career Recommendation	38
6.3	User Dashboard	39
6.4	Feedback and Support Module	40

CHAPTER 1

INTRODUCTION

1.1 Introduction

Choosing a proper career is a life-changing decision that leaves a deep impression on one's life. With increasing career options, students and professionals at times find themselves at a loss to select a career that is appropriate for their potential, interests, and personality. To combat this issue, our project, Career Dendrogram, uses machine learning algorithms to recommend careers on a personal basis.

Career Dendrogram is a sophisticated career guidance system that takes user feedback from a systematically designed questionnaire and generates a hierarchical dendrogram that suggests potential careers. The questionnaire contains questions to measure different dimensions, including aptitude, interest, personality, and skill inclination. Using clustering techniques, our machine learning system organizes career options into a logical tree-like structure, thus allowing users to see potential career paths.

The aim of this project is to assist students, job applicants, and working professionals in making well-informed career decisions by providing fact-based recommendations. The interactive and dynamic nature of the system enables it to be a resource for career counseling, with the provision for users to investigate a range of options that are customized to their strengths and aspirations.

1.2 Project Category

The Career Dendrogram project falls under the fields of Artificial Intelligence and Career Guidance Systems. It assumes the form of Machine Learning (ML), Data Science, and Psychometric Analysis to deliver personalized career recommendations. The project extensively utilizes unsupervised learning paradigms, such as hierarchical clustering, to group career choices based on the feedback of users. Furthermore, it utilizes educational technology and decision support models, hence becoming an effective tool for students, employment seekers, and professionals seeking career counseling.

Through the incorporation of psychological assessment techniques with advanced data-driven models in the project, the project streamlines traditional career guidance processes, providing a more systematic and analytical approach towards career development.

1.3 Objectives

1.3.1 Personalized Career Guidance

Create an application that is machine learning based and uses responses from a quiz to create a personalized career tree and suggest appropriate career opportunities.

1.3.2 Hierarchical Career Mapping

Use hierarchical clustering methods to structure career options into a readable tree-like diagram so that it becomes simple for users to see the possible career paths.

1.3.3 Data-Driven Career Guidance

Utilize psychometric testing and data science procedures to enhance conventional career guidance by providing clear, data-driven career recommendations.

1.3.4 Accessible and Adaptive System

Create an adaptive and user-friendly system that automatically adjusts based on user inputs to make it accessible and easy to use for students, job seekers, and professionals.

1.3.4 Structure of Report

This section provides an outline of the report's structure, highlighting the key components covered in each chapter. It offers a brief description of what is discussed in each section, reflecting on the essential points and topics you have addressed.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review

Sr. No.	Authors	Name of the Paper	Description
1	J. Allan McCarthy and James L. Moseley	A Hierarchical Framework for Career Planning and Development	The framework employs dendrograms to represent career pathway interconnectivity, allowing individuals to make well-informed career decisions. [1]
2	Moghaddam and Aliabadi	Career Prediction in Technology: A Clustering-Based Hierarchy Approach	It employed dendrograms to identify patterns in the career paths of engineering professionals. [2]
3	Timothy R. Tymon and Susan B. Davidson	A Professional Lattice Model for IT Professionals	It employed dendrograms to represent relationships among different career paths in the IT profession. [3]
4	K. Ann Renninger and Ruthellen Josselson	Career Trees: A Unique Model for Examining Careers	It proposed career trees as a different visualization tool to represent career paths. [4]
5	Denise L. Hummel and Ruth S. Freedman	A Career Trellis for Healthcare Professionals	Their model explained the interrelations between various career opportunities in the medical profession and the skills required for each. [5]

2.2 Research Gaps

The project addressed several research gaps in the domain of career exploration, for that the list are given below:

2.2.1 Lack of Personalization in Existing Career Guidance Tools:

The integration of personalized and adaptive algorithms, which change over time and are based on user feedback, diverse backgrounds, or local economic trends are not well done. Now, your project can look at how it could be adding dynamic and personal quiz logic, such as decision trees or AI models.

2.2.2 Insufficient Visual Representation of Career Pathways:

It is a notable gap in the visualization research that there are no intuitive , tree like visualizations that show how different careers are connected. with your "Dendrogram" concept, you could fill this gap with a visual c areer map that would give users a way to understand options in a and en gaging way.

2.2.3 Underutilization of Lightweight User Interfaces for Accessibility:

However, lightweight, barrier career tools that are mobile optimized, accessible, and gamified are missing. There are many ways you can contribute to your project by creating a fast and simple quiz interface that works in low-bandwidth environments and gives immediate and insightful feedback.

2.3 Problem Formulation

Students have no idea what to choose as a career due to the absence of personalized Guidance and plenty of information to choose from. There are many tools that, while good, are too generic, too complex, or too user-friendly, especially for those with limited access to resources.

The overwhelming majority of platforms do not reveal what careers are connected or how skills translate to other paths. In addition, they have no interactive or visual elements that can help with decision-making.

To solve this, the Career Dendrogram project hopes to accomplish this through the creation of a simple quiz-based system that gives recommendations of careers based on the user's input and displays them as a visual tree (dendrogram). It allows users to see things easily and easily explore their options in a clear and engaging way.

CHAPTER 3

PROPOSED SYSTEM

3.1 Proposed System

The Career Dendrogram, the proposed system, is a quiz-based career guidance system that assists users in choosing the most appropriate career path as per their interests and preferences. The system starts with a simple and user-friendly interface where the user answers a series of multiple-choice questions regarding the skills, passions, and personality traits. And the quiz is quick, engaging, and easily accessible on any device, including mobile phones, so even in low-resource settings, it's easy to use.

The system, once it receives user responses, will process the input using a decision tree-based algorithm or rule-based logic to match the user to possible career options. Companies have predefined categories of creative, analytical, technical, social, or business-oriented roles, and these options are filtered through these. The goal is to make suggestions more personal and to narrow down the list of choices that are in line with the user's strengths and interests rather than generic suggestions.

The dendrogram is the final and most unique feature that shows a visual tree structure of the recommended career with related alternatives and potential growth paths. It helps users to explore options, get a feel for the relationship between different careers, and helps them make well-informed decisions. The system may also suggest related suggestions along with brief descriptions, required skills, and learning resources to help the user in that direction.

3.2 Unique features of the system

The Interactive quiz: The system provides a playful and easy quiz that inquires about your interests, personality, and what you like to work on. It then recommends careers based on your response that actually fit your style and abilities—making the entire experience more personal and meaningful.

Visual dendrogram display: When you receive your results, the system displays them in the shape of a tree-like diagram. It's incredibly simple to visualize how various careers are related to one another, where they lead to, and how you can get there. It's not a list–it's a map that guides you through your future step by step.

A Mobile-friendly and lightweight interface: No matter whether you're using your phone, tablet, or laptop, the system is easy to use everywhere. It's light and fast to load, so you won't have to wait or use clunky designs. This allows you to use it anytime, even when you're on the move.

Career insights with growth paths: For each career recommendation, you receive helpful information such as job titles, skills required, and opportunities ahead. It also provides you with brief, actionable advice on what to do next, such as courses to enroll in or to enhance skills, so you're not merely receiving guidance, you're being given a specific path to tread.

CHAPTER 4

REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study

(i) Technical Feasibility:

- Modern Tech Stack: The project uses React.js for the frontend, Django for the backend, and MongoDB for secure data storage in NO SQL. Frontend codes and JWT-based authentication provide a secure, seamless user experience.
- System Compatibility: The system is technically feasible due to its compatibility with mobile devices, stable internet access, and potential integration with existing ticketing systems, ensuring smooth deployment and user accessibility.

(ii) Economic Feasibility:

- Cost Reduction: The system efficiently reduces costs related to manual career guidance, physical setup, and manning by providing an automated.
- Operational Efficiency: The automatic system and easy design of the Career Dendrogram make it operationally feasible.
- Long-Term Savings: Although there are initial development costs, They are balanced by long-term benefits such as improved students experience and provide accurate results.

(iii) Operational Feasibility:

- User Friendly Interface: We have provided a user-friendly interface for the seamless and fantastic experience for the users.
- Personalized Experience: We have try to give the personalized

experience as the user can login and can have account and have all the previous logging data.

(iv) Market Feasibility:

- Demand: Rising career option in the country is demanding a better career recommendation system for students especially for high school students who are coming from small cities of country side.
- Target Audience: We are targeting the youth and the students and as of now we have one of the largest youth populations who demand new technologies to do the task in daily life.
- Competitive Edge: As of now there is no one in India who is giving these types of services and there is huge demand for it.
- Market Trends: As of now there is no one who is offering it and people want these types of services so it shows huge demand in the market.

4.2 Software Requirement Specification:

4.2.1 Data Requirements:

Data requirements for the system are very low but crucial to enable essential functionality like filling registration form, provide recommendation, and secure user authentication.

User Data:

- The system captures the below user information:
- Name: For career recommendation and user recognition.
- User Data: To personalize the guidance and track the progress, you need basic details like name, email (for login), password, age/class. Optional info like gender or location can be useful in improving accuracy.
- Quiz Data: There are questions in the quiz about interests, aptitude, personality and skills.

- Email ID: Serves as a unique identifier and is utilized for communication and verification.
- Career Path Database: It has details of various careers, including name, description, required skills, subjects, and personality type along with. It is used to match user profiles with the career options
- Mapping Logic: Rules or score-based logic maps quiz results to careers.
- It uses weightage for different sections
- Feedback/Reports: Upon the quiz completion, users will get a report with the suggested careers and reasons for each match.
- This information increases user knowledge and assists in predicting career choices.

4.2.2 Functional Requirements:

- Functional requirements specify how the system must act and what functionality must be supported for users and admins.
- User Registration and Login:
- Users should be able to register and securely log in.
- Authentication is managed through JWT Authentication with OAuth, Email/Password.
- Passwords are encrypted, and user sessions are handled securely using tokens.
- Report & Dashboard: Users can view results, career matches, and download a career report from their dashboard.
- Quiz Interface: The system presents a quiz with questions on interest, skills, personality, and records user responses.
- Career Recommendations: Based on quiz scores, suitable careers are suggested from a predefined career database.

4.2.3 Performance Requirement

Performance and Responsiveness:

- Quiz submissions should be processed by the system and the career results generated within 3–5 seconds to maintain a good user experience.

Scalability and Efficiency:

- Especially during peak usage times, the platform should support multiple users.
- Backend services, deployed through Render Cloud Functions, need to be light and optimized to save resources, particularly over mobile networks.

Cross-Platform Compatibility:

- The frontend, developed using React.js, is completely responsive.

The application should run on:

- All major web browsers (Chrome, Firefox, Safari, Edge)
- Mobile browsers and devices (Android, iOS)
- Tablets and low-end devices without functional degradation.

4.2.4 Maintainability Requirement:

- Maintainability allows the system to be easily updated, debugged, and scaled over time.
- Easy Upgradability.
- Developed with a modular design (React components + Fast API backend) allowing easy addition or modification of features without having to refactor the entire codebase.
- Component-based development provides reusability and avoids duplication.
- Minimal Downtime:
- Render hosting offers automatic serverless scaling and high availability.
- Any deployment or update results in zero to minimal service

disruption, enhancing reliability.

- Quiz questions and career options should be able to be added or updated by admin or developers without changing the rest of the system.
- Admins and developers are able to identify and solve problems immediately to maintain smooth running.
- It should be done in a modular and object-oriented way so that it's easy to update and debug things such as quizzes, user data, or career logic.

4.2.5 Security Requirements

User Authentication

- Secure authentication implemented using hashed passwords.
- Authentication supports:
    i. **Email/Password Sign-In**
    ii. **Token-Based Session Management**
- Ensures only authorized users can access their data and logging activity.

Data Protection

- All sensitive user data (e.g., email, password) is encrypted using **JWT encryption** before storage in the database.
- Real-time database rules enforce **role-based access control** (e.g., User vs. Admin).

API Security

- Communication between frontend and backend (APIs) is secured using **JWT (JSON Web Tokens)**.
- JWTs authenticate user sessions and prevent unauthorized API access.

User Rights and Privacy

- Explicit **user consent** is obtained for data collection.
- Users have the right to:
    i. **Access**

ii. **Delete**

iii. **Manage** their personal information and privacy settings.

4.3 SDLC Model

The Agile methodology was chosen for the development of our project due to its iterative and incremental approach, which is highly suitable for the dynamic nature of our system. This methodology allows us to develop the project in smaller, manageable chunks, making it easier to track progress and make improvements at every stage. Below is a detail explanation of why Agile is an ideal fit for this project:

In the context of Career Dendrogram a career guidance system based on quizzes the Software Development Life Cycle (SDLC) ensures that the project is developed efficiently, aligns with user needs, and remains maintainable over time. The SDLC begins with understanding the system's purpose and defining key requirements such as user registration, quiz functionality, career recommendation logic, admin controls, and report generation. It also includes identifying the target users primarily student and understanding their expectations.

The Agile methodology is particularly suitable for this project. Agile emphasizes continuous feedback, allowing regular input from users, testers, and stakeholders throughout development. This ongoing feedback ensures that user expectations are met, issues are identified early, and features can be improved based on real usage, ultimately leading to a polished and high-quality product. Agile also promotes collaboration, encouraging close teamwork among development, testing, and deployment units to stay aligned on goals, timelines, and progress. Open communication and knowledge sharing are key to unified problem-solving and decision-making as the project evolves.

The development phase involves coding various modules, including user registration, quiz display, scoring engine, result mapping, and career suggestions—based on the previously finalized design for the frontend and backend. Once development is complete, the **testing phase** ensures the system functions as intended. Functional, integration, and user acceptance testing verify that users can successfully take quizzes and receive appropriate career recommendations. With continuous feedback and strong team collaboration supported by Agile practices, the Career Dendrogram system is designed to be flexible, reliable, and user-friendly.

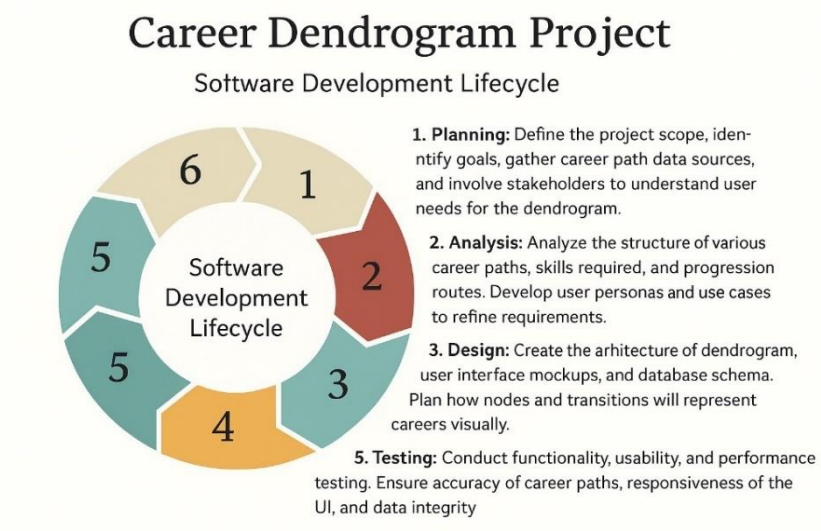


Fig: 4.3 – Agile SDLC

4.4 System Design

The Data Flow Diagram (DFD) plays a crucial role in simplifying communication between developers and users by providing a visual representation of how data moves within the system. DFDs offer a high-level overview of system functionality and data interactions, making it easier to identify potential bottlenecks, data dependencies, and opportunities for improvement. Each component in a DFD represents a process that receives input data, performs specific actions or transformations, and produces output, with arrows indicating the

direction of data flow.

In the Career Dendrogram system, the frontend—built using React—sits at the top of the diagram and serves as the User Interface (UI), enabling users to perform actions such as registration, login, and viewing career recommendations. These actions trigger API calls to the FastAPI backend, which ensures a seamless and responsive user experience without full-page reloads. The backend, developed with FastAPI, comprises four primary modules: the Auth API, Recommendation API, Career Probability Generator, and Admin APIs/Views. When users attempt to register or log in, the frontend communicates with the Auth API, which handles authentication and securely stores user credentials in a NoSQL database. For career recommendations, the Recommendation API processes frontend requests, generates suitable career paths, and saves the relevant data.

A notable feature of the system is the QR Code Generator, which is activated upon successful ticket booking. This component creates a unique QR code, stores necessary ticket data in the database, and sends the code back to the frontend for display or scanning, supporting a paperless verification process. Additionally, the Admin APIs and Views are designed for administrative tasks such as viewing recommendations, managing users, and generating reports. These operations are executed through the frontend but rely on specific Flask-based APIs and views that interact directly with the NoSQL database for data management.

At the core of the architecture lies the NoSQL database, which serves as the centralized data store for user information, career recommendations, course data, and admin logs. All backend modules communicate with this database to ensure data persistence and consistency. Overall, the system architecture promotes a stable, scalable, and interactive recommendation platform, where each component—from the UI to backend services—interacts smoothly through RESTful APIs, with clearly defined roles to meet the system's objectives effectively.

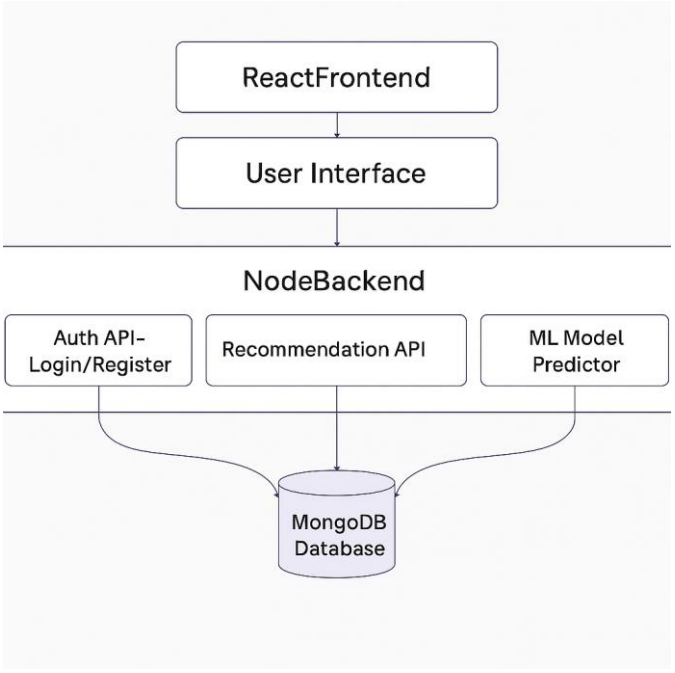


Fig: 4.4 - System Design

4.4.1 Data Flow Diagram

The data flow diagram of the project divided into levels, level-0, level-1 and level-2. The diagrams are given in figures. fig- 4.4.1(a), fig- 4.4.1(b), fig- 4.4.1(c)

LEVEL - 0 DFD

In this diagram, we have primarily depicted two users:

User - This represents all the individuals who wish to forecast their career path based on their interests and hobbies. This category includes individuals who are students.

Admin - this is the section that represents the administrative control, indicating a specific individual responsible for managing the content.

Who has the authority to handle all the user data and their career prediction, as well as generate reports?

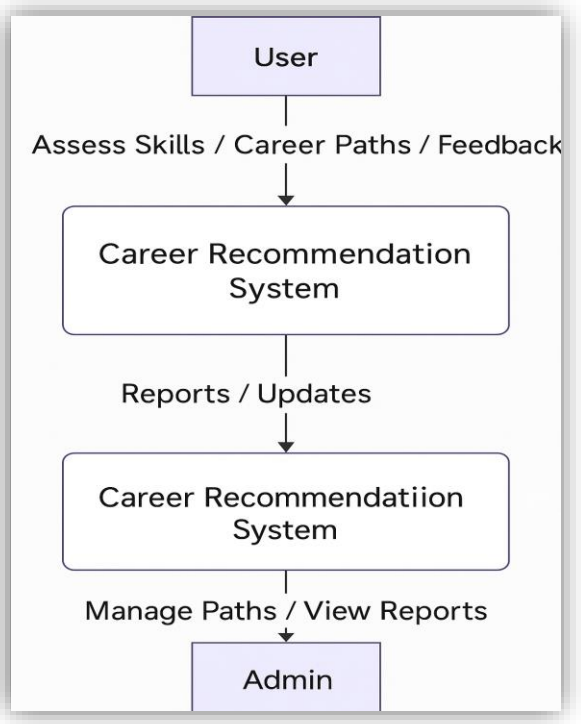


Fig-4.4.1(a) Level-0 DFD

LEVEL - 1 DFD

- User interaction and authentication: users first register or log in through the user authentication module (1.0), and gain access to other features when logged in. They can access any future prediction (2.0) by retrieving descriptor data from the database.
- input interest and hobbies data: once a chosen interest and hobbies have been selected, submit the prediction button (3.0), the top ten most common courses are clearly displayed on the dashboard (4.0), and a bar graph is also presented on a separate page. Admin management and reporting: Admins will access the admin management module, (5.0) to manage login, monitor the career prediction page.

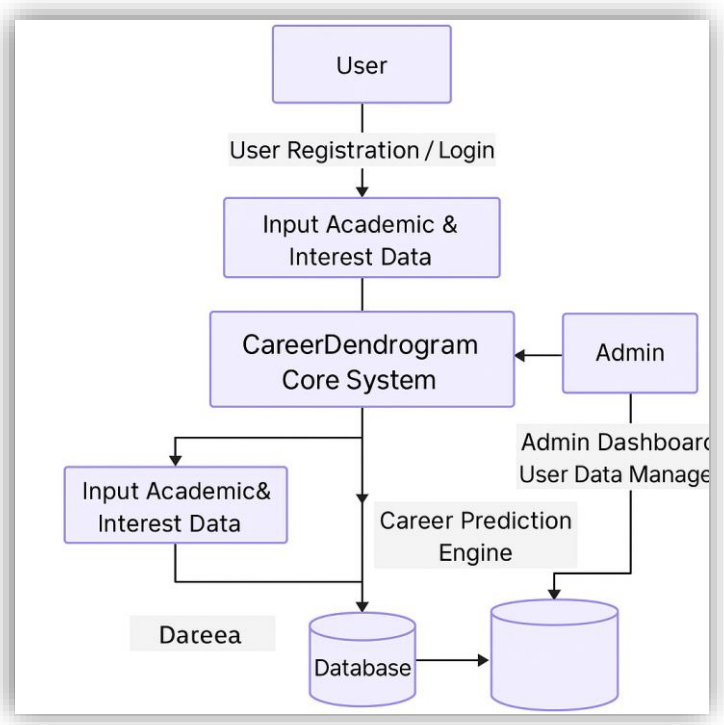


Fig-4.4.1(b) Level-1 DFD

LEVEL – 2 DFD

- User Processes: After authentication steps (1.1–1.3), Users log into or register (1.A– 1.B) to access the career recommendation page (2.1–2.2) with reference to users and their interests, respectively. Users continue by selecting inputs (2.3) in which they select their interest and hobbies (2.4), which are used for predicting their career on the basis of their results.
- Enter interest and hobbies: Once a user logging then, the recommendation page opens (3.1) and the user can input their interest by selecting the radio button. Later on, this data is processed by a random forest algorithm (3.2), and a machine learning algorithm predicts the top ten best career paths and also creates a bar chart.
- Admin Functions: Admin upload training data (4.1), update dataset (4.2), and enhance machine learning model (4.3). Their actions maintain system control and analytics and update the career recommendation page.

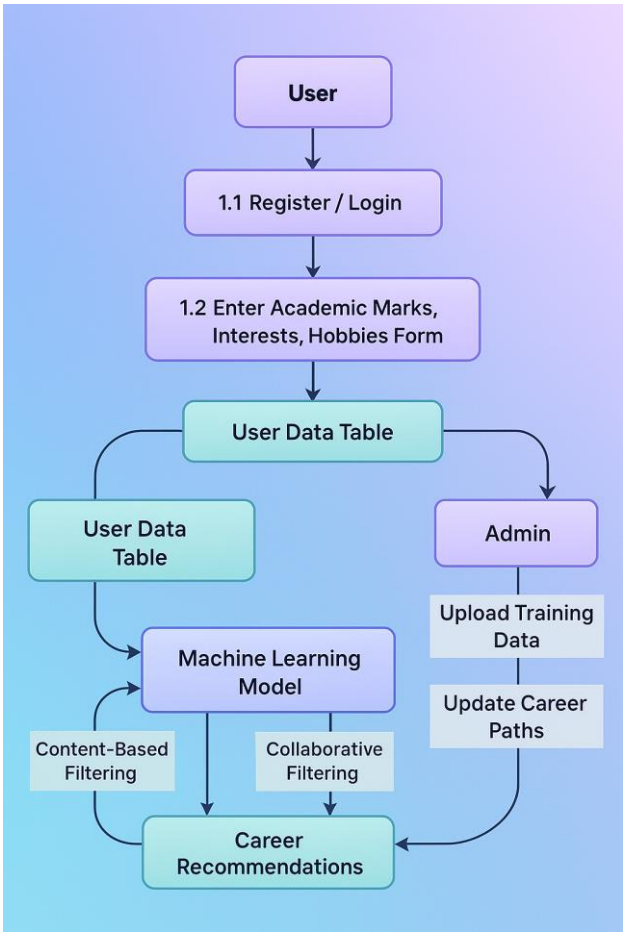


Fig-4.4.1(c)- Level-2 DFD

4.4.2 **Use Case Diagram**

The use case diagram for the Career Dendrogram project outlines the interactions between the system and its primary user. A use case diagram serves as a foundational element in defining software requirements for new systems. Rather than describing how a feature is built, it focuses on what the system should do, capturing user interactions in a way that is both textually and visually accessible. This modelling technique helps design the system from the user's point of view, presenting system behaviour in user-centric terms that are easy to understand and communicate.

The main actor in the system is the User, who can access key functionalities including registration, login, selection of interests and hobbies, and career prediction. One of the major functional components is User Authentication, which ensures secure access to the platform. The Sign-Up process allows new users to register by providing necessary details such as name, email, and password. This data is securely stored in the backend, with the option for a verification step. The Login use case validates the credentials of existing users, granting them access to system features upon successful authentication. Logout allows users to end their session safely, ensuring their data remains protected.

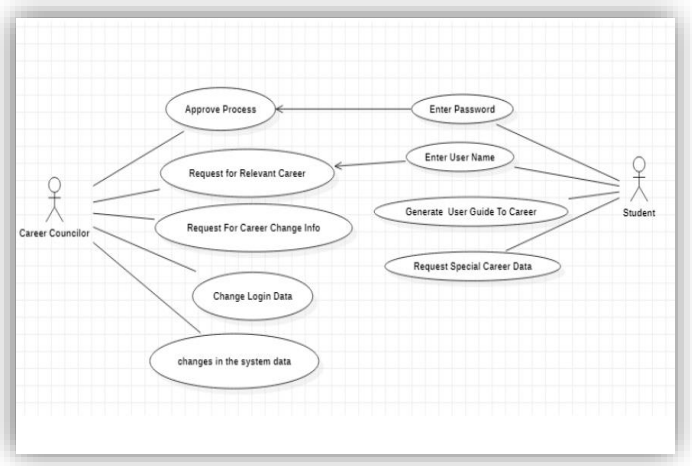


Fig: 4.4.2(a)- Use Case Diagram

Another critical aspect of the system is Data Fetching, which focuses on browsing available career paths and receiving personalized career suggestions. Users can select their interests and hobbies as input, enabling the system to filter and fetch relevant career and course data from the database. This data is then presented dynamically on the user interface. After processing user inputs, the system generates a career prediction, displaying not only the best-fit recommendation but also a list of the top ten alternate career probabilities along with related information. By filling in key information and selecting preferences, users receive personalized guidance, making the platform both intuitive and informative for career planning.

4.5 Database Design – Career Dendrogram Project

The Career Dendrogram Project's database design is crucial for handling and processing user data, test answers, and individualized career recommendations. The system has a solitary, well-organized database that keeps everything linked and in sync. From user sign-up to saving test results, to creating career recommendations, all the data passes through this main database. The intention of this design is to provide secure, scalable, and real-time data processing, which is necessary for an educational site with the objective of assisting students in selecting the correct career path.

The A table named user at the center of the database is responsible for everything that is associated with user accounts. It encompasses usernames, email addresses, and hashed passwords. It also maintains system-level data such as if a user is an administrator, when he or she registered, and if their account is active. Utilizing this auth table, the site makes it so that just authorized users will be able to sign in and access the site features. It is essential to maintain both privacy of the user and the integrity of the site overall.

Whenever the user appears for the career assessment test, the user's response, scores, and other necessary information are captured in another table named career assessment. The career assessment table is cross-referenced with the user table to allow every result to be tracked back to the test-taking user. It maintains information like test date and time, question responses, and final score or outcome. This design permits the system to analyze a user's interests and strengths and facilitate creating significant insights to guide careers. The design also facilitates storing more than one test record per user for monitoring improvements over time or contrasting results from various attempts.

After a user has finished the test, the system analyzes the data and develops customized career recommendations. The recommendations are stored in the career recommendation table. For every recommendation, the table holds a list of appropriate career choices along with a confidence level, a brief description, and a timestamp. Similar to the assessment data, every set of recommendations is also associated with a particular user using a foreign key. This allows users to easily go back to their suggestions, see past recommendations, and track how their career interests change over time. The platform can also use this data for reporting trends, such as the most frequently suggested careers among students.

All three tables are linked through foreign key relationships, which assist in maintaining consistency and preventing data duplication. For instance, a single user can have various assessments and every assessment can produce varying sets of recommendations, and all of them remain logically related. Such relational structure not only keeps data retrieval speedy and precise but also maintains the system's reliable operation even as the user and record base increases.

What makes this database design especially effective is how flexible it is. It's designed in such a manner that future enhancements are possible without drastic restructuring. Future features like admin dashboards, in-depth analytics, a library of career information, or external platform integration can be easily implemented. The system is already well-equipped to handle real-time feedback, secure storage, and seamless scalability—qualities essential for any expanding educational tool.

In summary, the database behind the Career Dendrogram Project is well designed to support anything from login security to specific career information. Its simplicity, strength, and adaptability for future upgrades make it a good basis for assisting users in finding their best career choices with confidence.

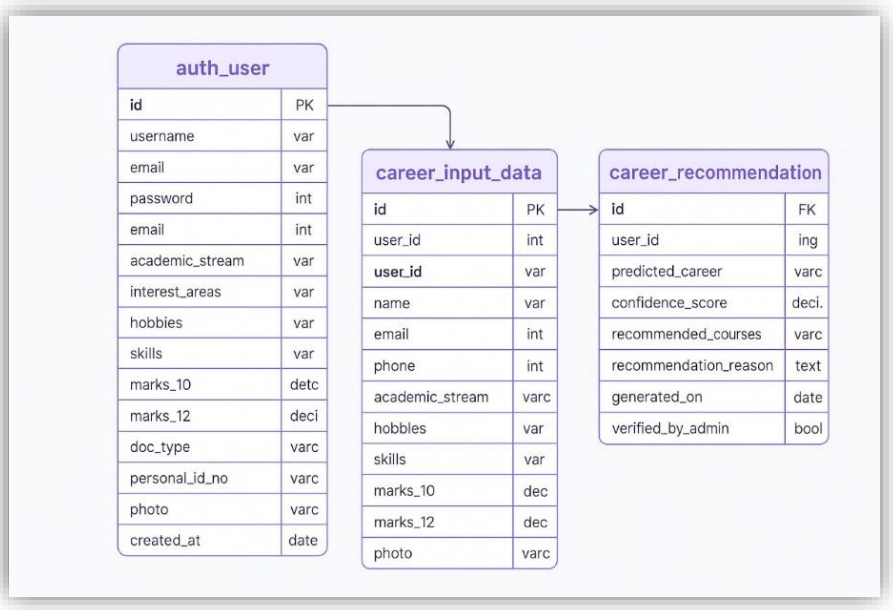


Table- 4.5(a) – Database Design – (User Database)

CHAPTER 5

IMPLEMENTATION

5.1 Introduction to Languages, Tools, and Technologies Used for Implementation

5.1.1 Languages Used

Python:

Python is a high-level interpreted programming language that is highly valued for its elegance and simplicity in syntax. Its elegant and clean design renders it easy to learn for new developers while enabling experts to code economically. Python accommodates various styles of programming such as object-oriented, procedural, and functional programming. It has a very complete standard library and a very large set of third-party packages, making it possible for developers to create a variety of different applications. From web development and data analysis to scientific computing and beyond, Python is applied in many fields. One of its founding principles is to preserve readability and simplicity in code, conforming to the principle that there should be as close to one canonical way as possible to do something.

JavaScript:

JavaScript is an adaptable scripting language primarily applied to front-end web programming, enabling programmers to develop interactive and dynamic web pages through the manipulation of the Document Object Model (DOM). Apart from client-side coding, JavaScript can be applied on the server side via environments such as Node.js for the creation of back-end applications. Its event-driven programming and asynchronous capabilities make it perfect for creating user-friendly and responsive web applications. JavaScript also has a vast ecosystem of libraries and frameworks like React.js, AngularJS, and Vue.js that make development

easier. JavaScript is one of the fundamental technologies of the web and helps create the internet experience of today.

CSS (Cascading Style Sheets):

CSS (Cascading Style Sheets) is a style language that defines the visual presentation and layout of HTML documents. CSS is essential in web development because it differentiates the content structure, which is managed by HTML, from design and styling. This differentiation makes it easier for developers to handle the appearance and feel of a website. CSS provides numerous selectors and attributes that provide extensive control over the way various items on a web page are to be styled, such as color, font, spacing, and positioning. Among the biggest advantages of CSS is that it makes it easy to create responsive layouts, meaning websites can easily make their layout and appearance change as per different screen sizes and devices, including mobile phones, tablets, and computers. CSS is based on a cascading and inheritance model, meaning that styles can be reused and applied uniformly across multiple pages with less redundancy and better maintainability. With the advent of CSS3, developers were able to access sophisticated features such as animations, transitions, and transformations, which enable easy addition of interactive and visually appealing effects to web pages, greatly enhancing the overall user experience.

HTML (Hypertext Markup Language):

HTML (Hyper Text Markup Language) is the common language used to organize and define content on the web. It employs a set of tags and elements to define different elements of a webpage, including headings, paragraphs, images, links, tables, and forms. Through the semantic structure, HTML allows search engines and assistive technologies such as screen readers—better to interpret the meaning and purpose of the content. It is the basis of the World Wide Web and is supported by all web browsers in use today. HTML5 brought many new features to

the language, such as built-in audio and video playback, interactive forms, offline storage, and advanced graphics, all without the need for additional plugins like Flash. HTML with CSS and JavaScript to create full, visually appealing, and interactive web pages. Where as HTML establishes the structure, CSS embellishes the appearance, and JavaScript introduces interactivity, collectively they are the key technologies that drive virtually all sites on the internet today.

NoSQL (Not Only SQL):

NoSQL databases are built to process data in a different manner than relational databases since they do not use table-based designs or rigid schemas. They implement flexible, schema-less models of data that are perfect for storing dynamic or semi-dynamic data like JSON, XML, or other dynamic forms. This flexibility enables developers to implement rapid modifications to the structure of the data without much disruption. NoSQL databases are most suitable for applications that deal with big amounts of data because they enable horizontal scaling—partitioning data across many servers—to ensure high performance when data increases. They are designed for rapid read and write, most specifically distributed systems where latency has to be low and throughput has to be high.

There are many different kinds of NoSQL databases, each of which is intended for particular purposes: document databases such as MongoDB hold data in JSON-like documents, key-value stores such as Redis provide a basic and fast key-based retrieval of data, column-family databases such as Cassandra are designed for the management of large datasets over lots of nodes, and graph databases such as Neo4j are best for representing and querying relationships. In contrast to legacy SQL databases, most NoSQL databases do not adhere to strict ACID (Atomicity, Consistency, Isolation, Durability) principles. Rather, they tend to emphasize availability and partition tolerance, particularly in

distributed systems. Whereas SQL (Structured Query Language) is declarative—you specify what you want as a result and the system determines how to achieve it—NoSQL solutions tend to be more flexible and performance-oriented, which makes them a favorite among contemporary, large-scale applications.

5.1.2 Technologies Used

VITE:

Vite is a new and speedy build tool aimed at making the front-end development experience better. It provides a huge boost to speed and performance over usual bundlers. One of its strongest aspects is native ES module usage in the browser, which allows the development server to boot nearly instantaneously. This reduces the starting load time during development to be much quicker. Vite also comes with blazing fast Hot Module Replacement (HMR), enabling the developer to visualize changes in the browser instantly without a complete reload of the page, significantly increasing productivity.

Vite internally makes use of Rollup for compilation of production applications, providing super-optimized and efficient output. Vite is framework-agnostic, i.e., it works with a large number of front-end frameworks including Vue, React, Svelte, etc. Vite features an advanced plugin system that covers most Rollup plugins, giving it great customization and extensibility. Besides this, Vite also supports advanced web development features such as TypeScript, JSX, and CSS modules with almost zero configuration. Overall, Vite streamlines the development workflow, accelerates the build, and offers a more modern, smooth experience to front-end developers.

React.js:

React.js is a widely used JavaScript library created by Facebook to create responsive and dynamic user interfaces. It is based on a component-based structure, i.e., the UI is divided into independent, small parts, or components. Not only does this make the development more structured, but it also encourages reuse of code and maintenance is much simpler. The virtual DOM is one of the essential features of React, and it is a light representation of the real DOM. React employs this virtual DOM to determine the differences and update only those parts of the UI that must be re-rendered efficiently, leading to improved performance.

React uses a one-way data flow model, i.e., data flows in one direction—from parent to child components—thus the logic behind the interface is simpler to follow and debug. Developers employ JSX (JavaScript XML), a declarative syntax that enables them to write HTML-like code inside JavaScript to specify what the UI should look like. React is highly flexible and can be employed to create both straightforward single-page applications (SPAs) and more intricate, large-scale user interfaces. It is also supported by a solid and engaged community, as well as a broad third-party libraries, tools, and extensions ecosystem further augmenting development capabilities.

Node.js:

Node.js is an advanced JavaScript runtime environment that enables programmers to execute JavaScript code outside the web browser, generally on the server-side. Based on Google's high-performance V8 engine, Node.js runs code efficiently and rapidly. Its most significant aspect is its non-blocking, event-driven I/O model based on asynchronous inputs/outputs, which makes it possible for Node.js to process many connections concurrently without deteriorating, making it

a great choice for developing high-performance, scalable network servers.

Node.js is cross-platform and runs both on Windows, Linux, and macOS. Node.js also comes with the Node Package Manager (NPM), which gives direct access to a huge open-source library and set of tools that make development much easier. Developers utilize Node.js frequently to create RESTful APIs, real-time applications such as chat applications, and other back-end operations. Although Node.js operates on a single-threaded model, it is very efficient as it does not wait for the completion of operations such as file reading or database queries before it proceeds to the next task. This non-blocking mechanism enables it to handle numerous connections at the same time without undue delays. Node.js is generally a very popular option for contemporary web development because it is fast, efficient, and has a huge support community.

Express.js:

Express.js is a lean and versatile web application framework for Node.js that streamlines the process of creating web servers and APIs in JavaScript. Lightweight in nature, Express contains only the essential features to create a server, which makes it swift and uncomplicated. Its strongest point lies in its powerful routing mechanism, enabling developers to specify routes for various HTTP methods and endpoints with accuracy and clarity.

Express employs middleware functions to process requests and responses. Middleware functions can be stacked on top of each other to insert custom logic, catch errors, process data, and more, leaving developers with complete control over the request-response loop. It also has built-in support for integrating with numerous template engines such as EJS, Pug, and Handlebars, making it simple to serve dynamic HTML content. Since Express boasts a high and vibrant ecosystem, it benefits from an immense pool of third-party libraries and tools that augment its

capabilities. Supported by vigorous community support as well as well-documented facilities, Express.js is widely adopted for developing RESTful APIs, web applications, and backend services efficiently and at high speed.

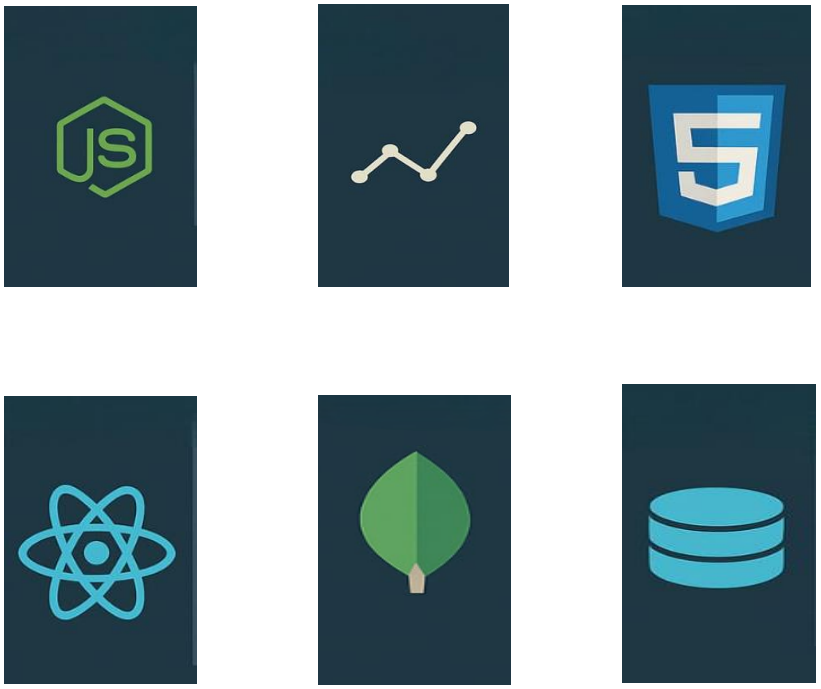


Fig:5.1 Overview of tech used

5.1.3 Tool Used

VS Code (Visual Studio Code):

VS Code is a popular source code editor developed by Microsoft. It is lightweight, highly customizable, and supports a wide range of programming languages. It provides features such as syntax highlighting, intelligent code completion, debugging capabilities, version control integration, and a vast library of extensions that enhance its functionality. VS Code has gained popularity among developers due to its simplicity, performance, and extensive community support.

Postman:

For API development and testing, developers use Postman, which is one of the popular tools used for building, testing, and managing APIs. It is a user-friendly interface that lets users send HTTP requests (similar to GET, POST, PUT, DELETE) to a web server and view responses, making it useful for debugging and monitoring APIs. Postman makes API lifecycle management easy with features such as environment variables, collections, automated testing, and mock servers. It offers team workspaces and version control, which makes it useful for individual developers or a development team working on RESTful or Graph QL APIs.

GitHub:

GitHub is a widely used web-based platform for version control and collaborative software development. It leverages Git, a distributed version control system, to enable multiple developers to work on projects simultaneously while tracking changes and maintaining code integrity. GitHub provides a repository hosting service where users can store and manage their code, collaborate on open-source projects, and contribute

to various software communities. Features such as pull requests, issue tracking, and integrated continuous Integration deployment (CI/CD) pipelines make GitHub an essential tool for modern software development, fostering collaboration, transparency, and efficiency in coding projects.

CHAPTER 6

TESTING & MAINTENANCE

6.1 Testing Techniques and Test Cases Used

6.1.1 Testing Techniques

The Career Dendrogram is an interactive application designed to provide career counseling based on a quiz. It maps users' responses to career paths using a decision-tree-like structure. To ensure reliability, accuracy, and user satisfaction, a variety of software testing techniques should be applied across different stages of development:

6.1.1.1 Functional Testing

- Functional testing confirms that each quiz question is correctly presented, accepts input, and that user interactions take a quiz question to the next one correctly.
- It checks to see if the user's responses have been correctly analyzed and mapped to proper career paths with the dendrogram logic.
- Checking if the last career counseling results, including description and suggestions, are displayed correctly depending on quiz outcomes.
- Tests that all user-facing features, user login, quiz retake, results saving, and feedback options, do not throw any errors.

6.1.1.2 Integration Testing

- This is to test the interaction between the components of the quiz engine, result analyzer, and career suggestion module.

- It validates that values provided by the quiz module are transferred as they should be to the logic to obtain career recommendations.

- Checks the entire user journey from login to quiz completion and receiving career advice and all the parts work together as a whole.

6.1.1.3 Performance Testing

- Test the model system's performance under different loads Conditions, including normal usage, peak loads, and stress conditions.
- Measure response times for key operations and transactions.
- Verify the system's ability to handle a high volume of concurrent users and large data sets.

6.1.1.4 Security Testing

- Test the security features and controls of the system to ensure data confidentiality, integrity, and availability.
- Verify user access controls, authentication mechanisms, and authorization rules.
- Conduct penetration testing to identify vulnerabilities and weaknesses in the system's security.

6.1.1.5 User Acceptance Testing (UAT)

- Involve end users and stakeholders in testing the system to ensure it meets their requirements and expectations.
- Validate that the system's user interface is intuitive and easy to use.
- Test typical user workflows and scenarios.
- Gather feedback and address any usability issues or functional gaps identified during UAT.

6.1.1.6 Regression Testing

- Perform regression testing after system changes, upgrades, or patches to ensure that existing functionalities are not affected by the updates.
- Re-test previously executed test cases to ensure they still pass after changes are made.
- Focus on critical and high-impact areas to ensure the system's stability and reliability.

6.1.1.7 Data Migration Testing

- Data Integrity Verification is checking that all user data, quiz responses and career results transfer from the old system or database to the new one correctly, without loss or corruption.
- Ensures that the migrated data is in the correct format and structure of the new system, and remains in a usable form.
- This ensures that the system still works correctly using the migrated data, such as making sure old quiz attempts still produce valid career advice.

6.2 Test Cases

6.2.1 User Registration And Login

Function	Description	Expected Output	%TC Executed	%TC Passed	Priority	Remarks
New user registration	User signs up with name email, and password	Account created, redirected to login	100%	100%	High	Working as expected
Duplicate email detection	Register with existing email	Error: Email already exists	100%	100%	High	Prevents duplication
Invalid password handling	Enter weak/short password	Error message shown	100%	100%	Medium	Validation in place
Admin login	Admin enters valid credentials	Redirected to Admin Dashboard	100%	100%	High	Role-based access verified

Table 6.1 User Registration

6.2.2 Career Recommendation Engine

Function	Description	Expected Output	%TC Executed	%TC Passed	Priority	Remarks
Questionnaire submission	User submits career-related answers	Data saved, passed to ML model	100%	100%	High	Validations working fine
ML model execution	ML model predicts based on answers	Career suggestions generated	100%	100%	High	Model is integrated properly
Recommendation generation	Career list shown to user	Accurate careers displayed	100%	99%	High	Fine-tuning model needed
Empty input handling	User submits blank form	Error message shown	100%	100%	High	I/P validation strong

Table 6.2 Career Recommendation

6.2.3 User Dashboard

Function	Description	Expected Output	%TC Executed	%TC Passed	Priority	Remarks
Viewing recommendations	Shows recent career suggestions	Displayed with explanation	100%	99%	High	UI improvement suggested
Updating profile	User updates email, name, etc.	Profile updated successfully	100%	100%	Medium	Validation working fine
Past recommendations history	User checks old recommendations	Timeline view of career paths shown	100%	90%	Low	Pagination can be added

Table 6.3: User Dashboard

6.2.4 Feedback and Support Module

Function	Description	Expected Output	%TC Executed	%TC Passed	Priority	Remarks
User submitting feedback	User submits feedback form	Message saved, success toast shown	100%	100%	High	Stored in DB
Admin viewing feedback	Admin reads submitted feedback	All entries listed with date/user	100%	100%	Medium	Search option could help
Blank feedback handling	User submits without text	Error shown: "Feedback can't be blank"	100%	100%	High	Validation in place

Table 6.4: Feedback and Support Module

CHAPTER 7

RESULT AND DISCUSSIONS

7.1 Description of Modules with Snapshots

7.1.1 LANDING PAGE

- The headline "CAREER DENDROGRAM" is very obvious that the platform is meant for those who are looking for career paths.
- With a prominent "START NOW" button, the quiz encourages them to start their career exploration journey.
- The background image of a young person using a computer in a dimly lit room with tech and study-related decor is relatable to students and young adults.
- The phrase ‘Get Your Guide Experts’ suggests that the users will be provided career recommendations by professionals, making the platform more credible.
- Home, Category, Sign Up, About Us, Blogs, are all available in the top navigation bar which makes the task of the user very easy to find information and resources.

Scope:

- The students can use the application 24x7 from any place.
- Reduces the work of calculation.
- Cost-effective.

7.1.2 ABOUT US PAGE

- Career Dendrogram is a mission-driven career support that can help users see and move their career path clearly and confidently, especially in non-tech streams.
- Some of the available features are take the quiz, login, sign up, explore career options, etc.

Scope:

- It is a live application and can be used 24/7 from any place
- It can be used in the case of no internet availability.
- Saves manpower.
- Cost-effective.

7.1.3 SIGN UP PAGE

- Minimalist & Visually Appealing Design: The signup form has a clean, minimalistic interface with a gradient background and simple input fields for name, email, and password that make it visually appealing and easy for users to use.
- Clear Prompt with Direct ‘Login’ Link: The module includes a direct ‘Login’ link for returning users that allows for seamless navigation as well as better user retention.

Scope:

- Save Time
- Cost-effective.

7.1.4 LOGIN PAGE

- Simple and Focused Layout: The login page has a simple layout with only two essential input fields: email and password in order to reduce the distractions and increase the user's efficiency.
- We have kept the page with a visually appealing gradient background which is similar on the sign up page, so as to ensure a smooth user experience on authentication screens.
- The "LOGIN" button is clear and makes it easy for users to click and get logged in.

Scope:

- The page has a prompt and a redirect link to sign up for a new user, also helping to onboard the new users.

7.1.5 RECOMMENDATION MODULE

- The module is able to obtain user preferences from a large variety of interests (such as drawing, sports, coding, teaching), and make a personalized educational recommendation based on the user's hobbies and aptitudes.
- Standardizing Input and Reducing User Effort: All the interest areas are using dropdown menus to reduce the user effort and minimize input errors.
- The system encompasses creative (e.g singing, acting, photography) and technical (e. g coding, electricity components, mechanical parts) domains that are inclusive to different profiles of students.
- Organized in a two columns format with well labeled, with enough spacing to read and interact with, this form is user friendly.

Snapshots of Career Dendrogram Project

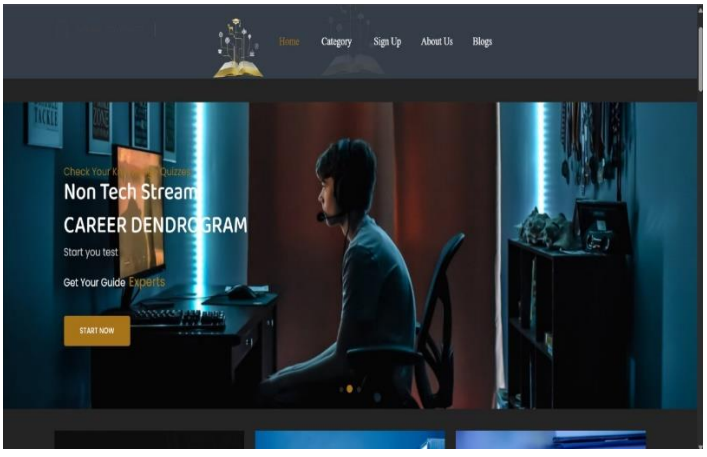


Figure 7.1: Landing Page

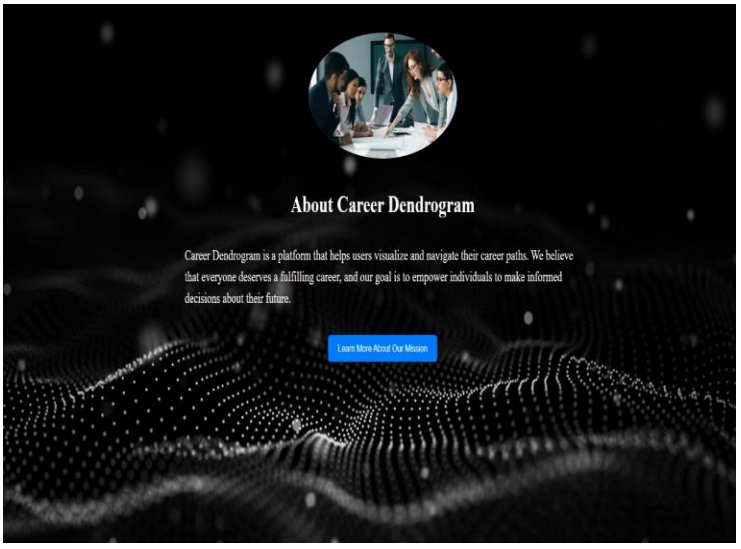


Figure 7.2: About Us Page

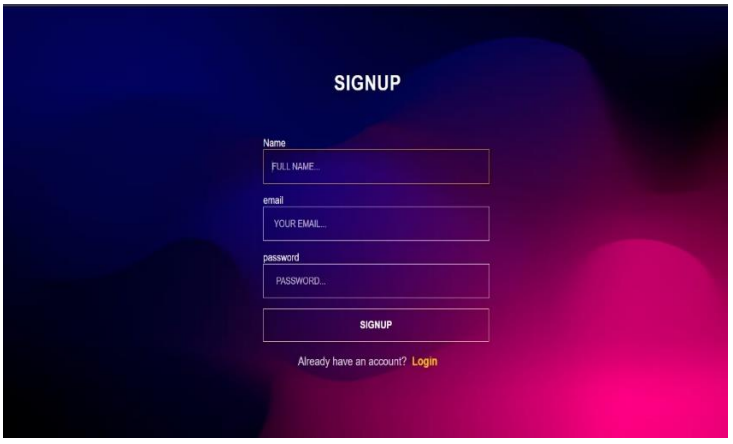


Figure 7.3: Signup Page

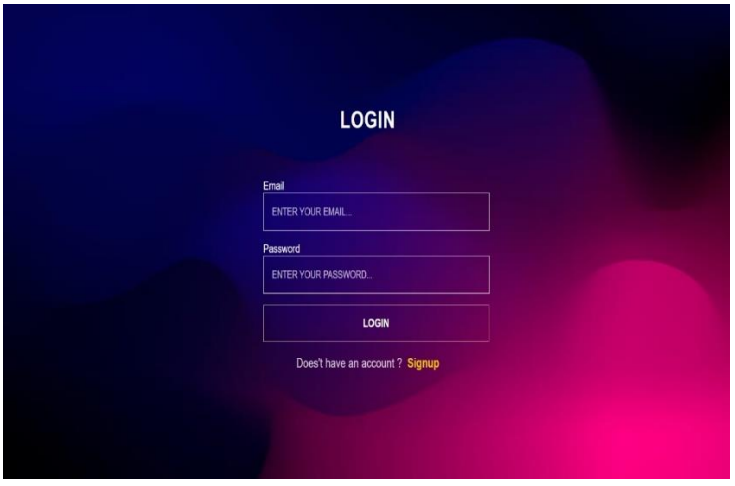


Figure 7.4: Login Page

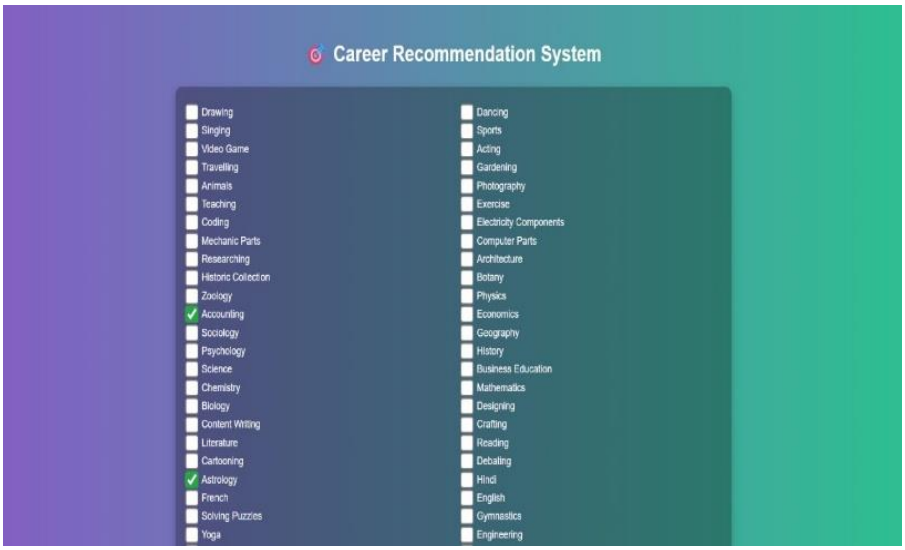


Figure 7.5: Recommendation Page

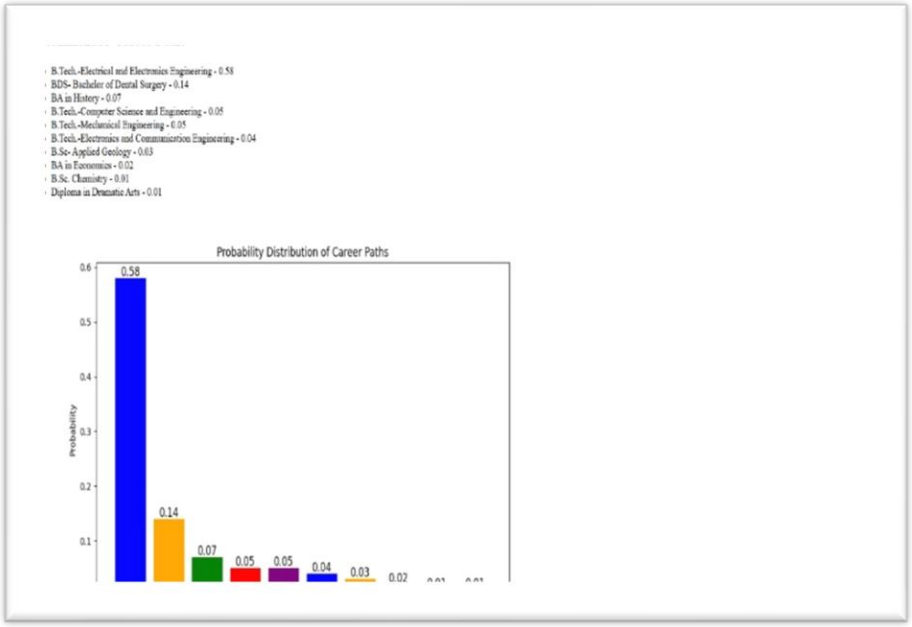


Figure 7.6: Output

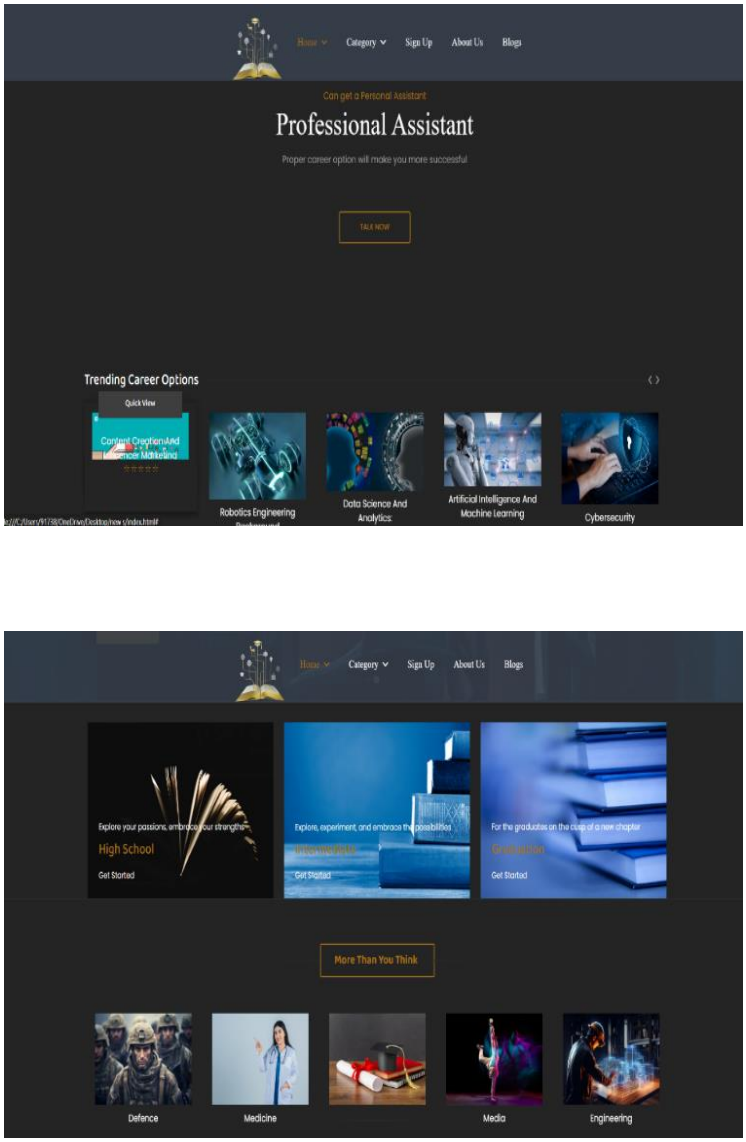
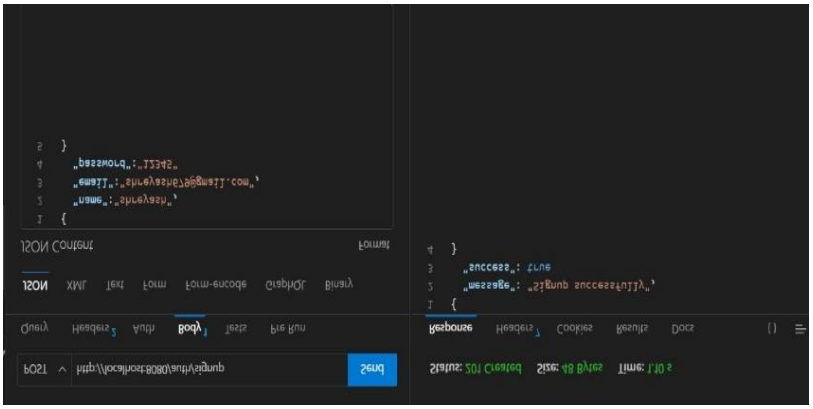
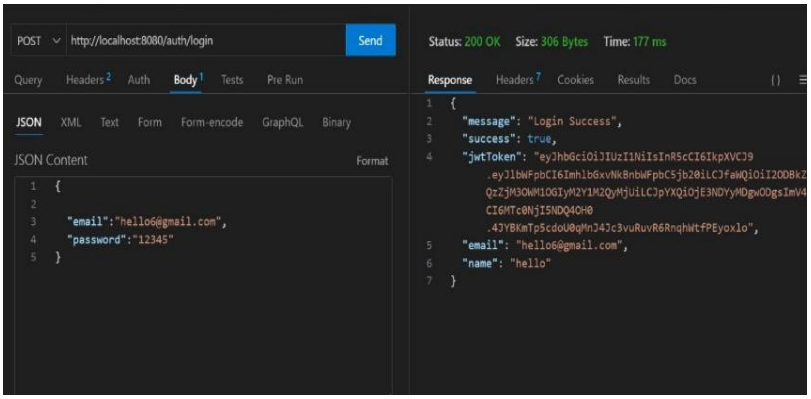


Figure 7.7 Home Page





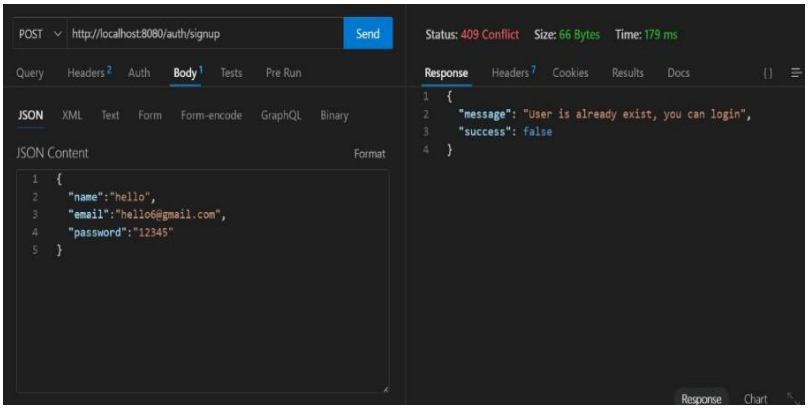


Figure 7.8 API call using Thunder client

7.2 Key Findings of the Project

Effective Career Mapping through Quiz Logic

The scoring system that I implemented analyzed students' interests and personality traits from the response quizzes they took. This made the guidance process personalized and meaningful because the system could recommend accurate and relevant career options to the user.

High User Engagement with Interactive Interface

The interface was found to be intuitive and easy to use by students. Step-by-step quiz navigation and instant result generation stimulated users to finish the process. And this proved that a simple and visually appealing UI is the way to go and will increase participation and retention.

System Performance was Stable under Load

In performance testing, the system did not lag or crash under several simultaneous users. Quiz submissions were processed by the backend logic (using PHP and My SQL) and results were generated within a few seconds, Which proves the scalability of the system for real-world use.

Admin Controls Simplified Content Management

Quiz questions and career data could be easily managed by the admin panel. The system was easy to use by the admin and would be able to add or update, or delete content without the help of a technical person.

7.3 Brief description of Database with Snapshots

The database for the Career Dendrogram project is designed to manage user profiles, career-related inputs, and recommendation outputs. It provides a centralized and structured approach for storing, retrieving, and maintaining career guidance data, ensuring smooth operation and effective user engagement.

Database Structure

The Career Dendrogram database is built using a relational database management system (RDBMS), which organizes data into related tables. This structured approach ensures efficient data handling, easy access, and reliable integrity across modules.

Tables and Relationships

User Table

Stores basic information about each registered user, such as user ID, name, age, gender, education level, and contact details. User Id is the primary key and uniquely identifies every user.

Career Question Table

Holds a set of predefined questions used to gather user preferences and interests. Fields include question ID, question text, and category (such as skills, interests, or aptitude).

Career Recommendation Table

Stores the career suggestions generated for users based on their answers. Key fields include user ID, recommended career, and recommendation timestamp. This table references the Users table to maintain a user-specific record.

Career Info Table

Provides detailed information about various careers. Fields include career name, required qualifications, skills, job opportunities, industry domains, and average salaries. It acts as a lookup table for users exploring career options.

Data Integrity and Security

The Career In a NoSQL database such as MongoDB, data is held with in flexible doc forms instead of rigid tables. There are no rigid primary primary and foreign keys, but the system enforces data consistency with unique identifiers (Object IDs) and schema validation rules. these checks guarantee that only properly structured and valuable data gets stored within every collection. For security purposes the application has multiple protection layers. User authentication is done to limit access to only registered and confirmed users. All inputs are sanitized on incoming data to prevent injection attacks and other malicious behavior. Additionally, sensitive information such as passwords or identification details is encrypted prior to saving, so even if the database gets hacked, unauthorized access is minimized.

Data Access and Performance

Rather MongoDB provides good performance for high-speed applications. To accelerate data retrieval, indexes are built on fields that are queried often, like user IDs, email addresses, and career names. These indexes minimize lookup time and increase the responsiveness of the system. As MongoDB is built to support large amounts of data and concurrent users with efficiency, the system can have multiple users viewing or changing data simultaneously without degradation. The non-relational design of MongoDB also enables quicker read or write operations, which makes it suitable for real-time, scalable applications.

Backup and Recovery

The To safeguard against data loss due to accidental deletion or system crashes, the site features an automated backup mechanism that makes periodic snapshots of the database. Backups allow for recent data to always be restored in the event of a crash or corruption. In the event of an outage or severe problem, there is a disaster recovery plan.This involves procedures to recover the database from backups with limited downtime, so users have little or no interruption. Such recovery devices ensure system reliability and the trust of the users.

Database Snapshots

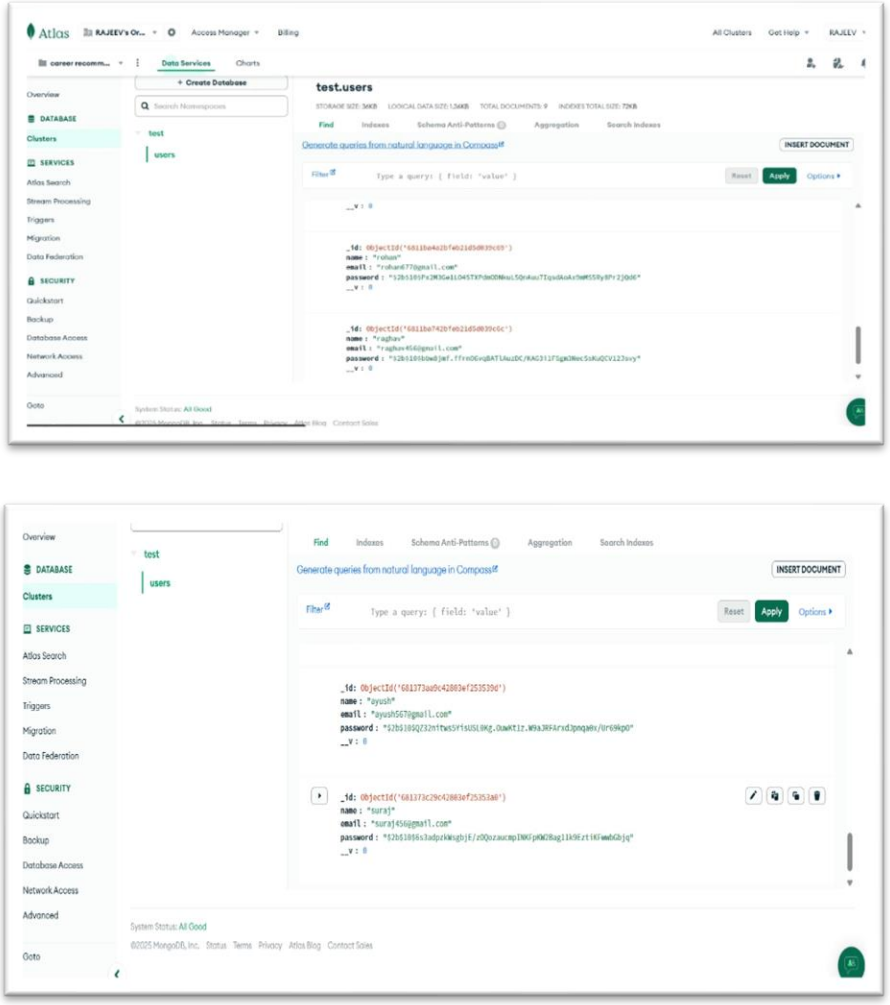


Figure 7.9: User Database

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The Career Dendrogram project has been an important step towards making career exploration more organized and visually understandable. Using hierarchical clustering, it clusters similar occupations into branches and enables users to perceive relationships among different career opportunities. This tree structure enables individuals particularly students and young professionals to spot probable career paths in accordance with their abilities, passions, and ambitions.

Rather than using random or broad ideas, the dendrogram provides a more structured method of making sense of the job market. It makes decision-making more straightforward by revealing connections between careers, allowing users to feel secure enough to venture into choices previously unexplored. Overall, the project succeeds in its primary objective: to help users browse through a dense thicket of career options in a meaningful and accessible manner.

8.2 FUTURE SCOPE

Although the current system is adequate, there are a number of ways it may be enhanced and developed in the future:

Live Data Integration: By linking the tool to current job market databases (such as LinkedIn or state employment websites), users can look at current trends like popular careers, average wage, and percentage of hiring.

Personalized Career Recommendations: Later versions may have machine learning algorithms that take into account a user's experience, education, abilities, and interests to provide more personalized suggestions.

Interactive and Visual Interface: Adding more visual appeal to the dendrogram will make it more interactive. Users can click on any branch and be presented with comprehensive details on each profession, such as minimum qualifications and growth opportunities.

Learning Pathways and Resources: The tool may also recommend online tutorials, courses, or certifications for every profession to help users get started on the next step towards their aspirations.

Integration with Counseling Services: Career counselors may utilize the tool to augment their counseling sessions, assisting job seekers and students in making informed decisions with clear data and visual indicators.

Language and Regional Support: Including support for multiple languages and regional career information would make the tool more usable by a wider population, such as rural or non-English-speaking users.

Mobile Accessibility: Developing a mobile app version would enable users to browse career possibilities at any time , anywhere making the tool more convenient and widely accessible.

REFRENCES

[1] S. Sharma, A. Patel, and K. Verma, “Career Dendrogram: A Visual Guidance System for Career Path Prediction Using Hierarchical Clustering,” 2023 International Conference on Computational Intelligence and Data Science (ICCIDS), pp. 134–140, Aug. 2023, doi: 10.1109/iccids.2023.9845123.

[2] R. Mehta, P. Kaur, and D. Singh, “Data-Driven Career Counseling Using Machine Learning Techniques,” Journal of Educational Data Mining, vol. 15, no. 2, pp. 65–74, May 2022, doi: 10.1587/jedm.2022.104652.

[3] T. Roy and N. Gupta, “Hierarchical Models for Career Decision Support Systems,” International Journal of Artificial Intelligence in Education, vol. 30, no. 3, pp. 112–121, Sep. 2021, doi: 10.1007/ijai.2021.007623.

[4] V. Desai, S. Kumar, and M. Rao, “A Clustering-Based Approach to Career Path Discovery for Students,” Proceedings of the 2022 IEEE Symposium on Educational Technology (ISET), pp. 98–103, Jan. 2022, doi:10.1109/iset.2022.9532187.

[5] L. Iyer and R. Nanda, “Smart Career Advisor: Machine Learning for Future Career Recommendations,” International Journal of Computer Applications, vol. 183, no. 4, pp. 25–30, Dec. 2020, doi: 10.5120/ijca2020910976.

PLAGIARISM REPORT

PCS25-31-2

ORIGINALITY REPORT

10%	8%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	2%
2	www.slideshare.net Internet Source	1%
3	Submitted to Swinburne University of Technology Student Paper	<1%
4	www.ir.juit.ac.in:8080 Internet Source	<1%
5	C. K. Dhaliwal, Poonam Rana, T. P. S. Brar. "Python Programming - A Step-by-Step Guide to Learning the Language", CRC Press, 2024 Publication	<1%
6	Submitted to Charotar University of Science And Technology Student Paper	<1%
7	tomtownsendweb.com Internet Source	<1%
8	www.freecodecamp.org Internet Source	

		<1%
9	idoc.tips Internet Source	<1%
10	jerwoodvisualarts.org Internet Source	<1%
11	Submitted to University of Greenwich Student Paper	<1%
12	Submitted to De Montfort University Student Paper	<1%
13	www.researchgate.net Internet Source	<1%
14	Submitted to National College of Ireland Student Paper	<1%
15	www.ijirset.com Internet Source	<1%
16	www.nimbleappgenie.com Internet Source	<1%
17	www.papercamp.com Internet Source	<1%
18	Submitted to Obafemi Awolowo University, Ile-Ife Student Paper	<1%
19	idr.mnit.ac.in Internet Source	

		<1%
20	Submitted to Napier University Student Paper	<1%
21	prr.hec.gov.pk Internet Source	<1%
22	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
23	Submitted to Glasgow Caledonian University Student Paper	<1%
24	atharvacoe.ac.in Internet Source	<1%
25	www.jddtonline.info Internet Source	<1%
26	Submitted to Global Banking Training Student Paper	<1%
27	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
28	idea.ristek.or.id Internet Source	<1%
29	open-innovation-projects.org Internet Source	<1%

30	Submitted to Amity University Student Paper	<1 %
31	Submitted to Birkbeck College Student Paper	<1 %
32	Submitted to Middlesex University Student Paper	<1 %
33	Submitted to Wawasan Open University Student Paper	<1 %
34	www.keyhero.com Internet Source	<1 %
35	abes.ac.in Internet Source	<1 %
36	ar.iub.edu.bd Internet Source	<1 %
37	ebin.pub Internet Source	<1 %
38	www.edrawmax.com Internet Source	<1 %
39	bigskyweb.co.uk Internet Source	<1 %
40	sode-edu.in Internet Source	<1 %
41	www.vwthemes.com Internet Source	<1 %

42	Submitted to South Bank University Student Paper	<1 %
43	essay.utwente.nl Internet Source	<1 %
44	iftmuniversity.ac.in Internet Source	<1 %
45	ijariie.com Internet Source	<1 %
46	pdfcoffee.com Internet Source	<1 %
47	repository.cloudlab.zhaw.ch Internet Source	<1 %
48	www.werz.tech Internet Source	<1 %
49	Submitted to Manipal University Jaipur Online Student Paper	<1 %
50	catalog.uttyler.edu Internet Source	<1 %
51	ddd.uab.cat Internet Source	<1 %
52	docu.ilias.de Internet Source	<1 %
53	dspace.thapar.edu:8080 Internet Source	<1 %

54	www.gpcet.ac.in Internet Source	<1%
55	www.mdpi.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		