

Career Pathway

Helping students in choosing higher studies

Project Report

Submitted by

Abhidarsh K S

Reg. No.: AJC23MCA-2002

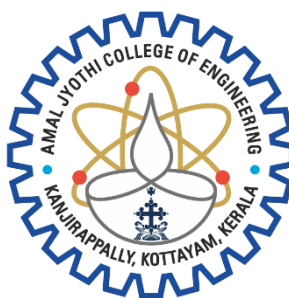
In Partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

(Accredited by NBA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

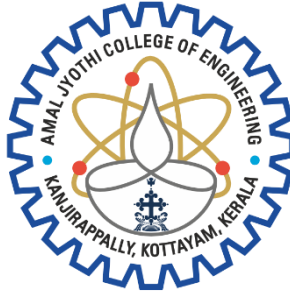


AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2025

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report titled “**Career Pathway**” is the bonafide work of **Abhidarsh K S (Regno: AJC23MCA-2002)** carried out in partial fulfillment of the requirements for the award of the **Degree of Master of Computer Applications** at **Amal Jyothi College of Engineering Autonomous, Kanjirappally**, Affiliated to **APJ Abdul Kalam Technological University**. The project was undertaken during the period from **December 10, 2024**, to **March 27, 2025**.

Ms. Merin Chacko
Internal Guide

Ms. Meera Rose Mathew
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

// CERTIFICATE ON PLAGIARISM CHECK

DECLARATION

I hereby declare that the project report “**Career Pathway**” is a bona fide work done at **Amal Jyothi College of Engineering Autonomous, Kanjirappally**, Affiliated to **APJ Abdul Kalam Technological University**, towards the partial fulfilment of the requirements for the award of the **Master of Computer Applications (MCA)** during the period from **December 10, 2024** to **March 27, 2025**.

Date:
KANJIRAPPALLY

Abhidarsh K S
Reg: AJC23MCA-2002

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director (Administration) **Rev. Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Merin Chacko** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

ABHIDARSH K S

ABSTRACT

Career Pathway is an intelligent educational and career guidance platform designed to help students navigate overwhelming academic and professional choices by offering personalized recommendations through a machine learning (ML) engine and provide a career path using AI. The system analyses user's educational backgrounds, psychometric test results (assessing quantitative, mathematical, and verbal skills), and career aspirations using AI and ML models like Decision Trees to suggest tailored career paths and courses that address skill gaps or align with strengths. It features modules for user, course, and job management, enabling administrators to curate listings while allowing students to explore career options, filter courses by category, and access job details. Additionally, the platform includes interactive elements such as educational blogs moderated by admins, user-generated content, and mock tests for skill assessment. By integrating data-driven insights with comprehensive resources, CareerPathway aims to streamline career decision-making while fostering a collaborative space for students and institutions to share knowledge.

CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	5
2.1	INTRODUCTION	6
2.2	EXISTING SYSTEM	6
2.2.1	DRAWBACKS OF EXISTING SYSTEM	8
2.3	PROPOSED SYSTEM	9
2.3.1	ADVANTAGES OF PROPOSED SYSTEM	10
3	REQUIREMENT ANALYSIS	12
3.1	FEASIBILITY STUDY	13
3.1.1	ECONOMICAL FEASIBILITY	13
3.1.2	TECHNICAL FEASIBILITY	14
3.1.3	BEHAVIORAL FEASIBILITY	14
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	15
3.1.5	GEOTAGGED PHOTOGRAPH	19
3.2	SYSTEM SPECIFICATION	21
3.2.1	HARDWARE SPECIFICATION	21
3.2.2	SOFTWARE SPECIFICATION	21
3.3	SOFTWARE DESCRIPTION	21
3.3.1	REACT JS	21
3.3.2	MONGO DB	22
4	SYSTEM DESIGN	24
4.1	INTRODUCTION	25
4.2	UML DIAGRAM	25
4.2.1	USE CASE DIAGRAM	26
4.2.2	SEQUENCE DIAGRAM	28
4.2.3	STATE CHART DIAGRAM	29
4.2.4	ACTIVITY DIAGRAM	30
4.2.5	CLASS DIAGRAM	33
4.2.6	OBJECT DIAGRAM	34

4.2.7	COMPONENT DIAGRAM	35
4.2.8	COLLABORATION DIAGRAM	36
4.2.8	DEPLOYMENT DIAGRAM	37
4.3	USER INTERFACE DESIGN USING FIGMA	39
4.4	DATABASE DESIGN	44
5	SYSTEM TESTING	52
5.1	INTRODUCTION	53
5.2	TEST PLAN	53
5.2.1	UNIT TESTING	54
5.2.2	INTEGRATION TESTING	54
5.2.3	VALIDATION TESTING	55
5.2.4	USER ACCEPTANCE TESTING	55
5.2.5	AUTOMATION TESTING	55
5.2.6	SELENIUM TESTING	56
6	IMPLEMENTATION	83
6.1	INTRODUCTION	84
6.2	IMPLEMENTATION PROCEDURE	84
6.2.1	USER TRAINING	84
6.2.2	TRAINING ON APPLICATION SOFTWARE	85
6.2.3	SYSTEM MAINTENANCE	85
6.2.4	HOSTING	85
7	CONCLUSION & FUTURE SCOPE	90
7.1	CONCLUSION	91
7.2	FUTURE SCOPE	91
8	BIBLIOGRAPHY	92
9	APPENDIX	94
9.1	SAMPLE CODE	96
9.2	SCREEN SHOTS	107
9.3	GIT LOG	112
9.4	CERTIFICATES	131
9.5	PLAGIARISM REPORT	134
9.6	PLAGIARISM AI REPORT	140

List of Abbreviations

- CP – Career Pathway
- UML - Unified Modelling Language
- ORM - Object-Relational Mapping
- MVT - Model-View-Template
- MVC - Model-View-Controller
- RDBMS - Relational Database Management System
- 1NF - First Normal Form
- 2NF - Second Normal Form
- 3NF - Third Normal Form
- IDE - Integrated Development Environment
- HTML - HyperText Markup Language
- JS - JavaScript
- JSX-JavaScript XML
- CSS - Cascading Style Sheets
- AJAX - Asynchronous JavaScript and XML
- JSON - JavaScript Object Notation
- API - Application Programming Interface
- UI - User Interface
- HTTP - Hypertext Transfer Protocol
- URL - Uniform Resource Locator
- PK - Primary Key
- FK - Foreign Key
- SQL - Structured Query Language
- CRUD - Create, Read, Update, Delete
- MERN-MongoDB, Express, React Js, Node Js

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

CareerPathway is a unique educational web platform aimed to offer the students, in general and those having passed out +2 or degree, an extensive set of resources and directions so they can take a better decision about their higher education and occupational trajectory. Adapting the challenges of the contemporary educational landscape, CareerPathway sets to close the information shortfall students encounter when picking courses or careers.

The platform aggregates all the structured and readily accessible data of courses, job opportunities, industry trends into a one-stop simplified UI. By putting this information together, CareerPathway hopes to calm the confusion and overstimulation of information that comes with the decision making for students.

1.2 PROJECT SPECIFICATION

CareerPathway platform is to enable students with comprehensive & user-friendly educational guidance system. Built for actual field-testing, the platform provides true value by empowering data-informed engagement with higher vs lower education options and careers.

User Roles and Functionalities:

1. Administrators:

- a. **User Management:** Administrators can control user accounts, supervise the activities and manage or enforce policies to keep secure environment for everyone.
Category Management: Admins can add / manage categories for courses and jobs thus making the system search organized to get all the relevant information.
- b. **Manager Management:** Administrators are able to create, update, and manage manager accounts in order to delegate responsibilities within the platform.
- c. **Report Management:** Admins review and fix user-reported blogs, with block any blog that is found to be inappropriate or Against community standards by the Administrators.
- d. **Manage Tests:** Admin adds/Deletes/manages the test, aptitude and personality test and aptitude time limit

2. Students (Users):

- a. **User Registration/Login:** Secure login and registration, that allow users to register with encrypted passwords once unique accounts that have been ..
- b. **Profile Management:** Users can be able to manage their profiles, view saved

content including courses and blogs, and personalise preferences for learning.

- c. **Course and Job Browsing:** Students can explore courses and careers by category with search filters that conveniently help to filter out options based on students likes and criteria.
- d. **Blog Access:** Authors of blogs and articles written by users, institutions or experts creates a collaborative learning space so users can read, comment on them.
- e. **Personalized Assessments:** Identifying strengths and development needs through assessments to inform program and career paths.
- f. **AI chat:** AI chat for answers of career and educational queries.
- g. **Personalized Recommendations:** System recommend career using ML of the aptitude and personality traits scores.
- h. **Provide Career Path:** By AI the system analyse the psychometric scores and educational background and guide him to do his dream career (Using AI to system analyse psychometric scores, Educational Background and gives guidance routes for those with their future job).

3. **Managers:**

- a. **Dynamic Course and Job Modules:** Managers are to add/edit courses information & job opportunities periodically so the content on platform is latest and relevant.

Conclusion

The **CareerPathway** platform was conceived as an advanced educational tool that supports what students struggle with when it comes selecting courses and careers. With its holistic resources, personalized perspectives tracking and collaborative functions, Career Pathway inspires students to reach their full potential in shaping the trajectory of higher and professional education.

Scopes Includes:

1. **Educational Guidance Platform:** They are proffesionals giving course and career specific information to students for academic & career planning.
2. **User Auth, Profile Setup:** Setup, edit and view privacy settings for students profiles (including keep track of everything they've saved, course / whatever are some examples)

3. **Course Job Module:** Offer to Admin and Managers where they can be continuously updating the regularly the course categories, JOBS or whatever details keep platform dynamic in current state.
4. **Blog Article Sharing:** Blog and article feed built into the system with user written posts, Institutional written publication or because of experts to share knowledge in a collaborative learning pattern.
5. **Career Path:** The students can measure or go through the exams and judge that which is the best career routes, and also what are skills and knowledge before this career.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The Pathway project is an experimental educational platform intended to provide +2 or graduate students the tool to make better decisions in the next phase of their higher studies and career. Built on cutting-edge web technologies, Pathway aims to give students an all-encompassing and easy experience in finding the best education opportunity for them; no more, no less.

As students face increasing challenges in selecting the right courses and career avenues, providing transparent and comprehensive data has never been more critical. This need is filled by Pathway, which provides users an online resource with courses and jobs they can browse, compare and understand at their own homes. It is shaped to cater for the various user needs — general users (students and institutions) as well managers/admins and support tools, that provide smooth in-depth experiences.

Key elements of Pathway are User Registration and Profile Management which provide user registration, with all associated scale-ups; detailed course and careers catalog along specifications for each course and career with appropriate description eligibility etc.; Real-Time Chat: Chat feature where users can communicate with each other in real time through the application.

Pathway also places value on shared learning, generating user posts from blogs and articles – which clarifying the creation of a social-learning space. One of the most distinctive features of Pathway is Psychometric Test — this is to help users understand their strengths and interests so they will be aligned with the right educational, career option. While this feature streamlines the decision-making process, it also supports a development and self-realization among students which calls the mission of the platform — to make student life easier.

Pathway hopes that by infusing these pieces, it can create a complete students navigation tool of higher education and career planning system to enable them to be successful and fulfilled in their future endeavors.

2.2 EXISTING SYSTEM

The current approach to career guidance and course selection is highly fragmented, relying on manual research, word-of-mouth recommendations, and scattered online resources. Students often struggle to find reliable and structured information about courses, job opportunities, and career paths, leading to confusion and uninformed decisions. Many existing platforms provide only general course details or job listings but lack an integrated approach that connects students' skills, interests, and career goals. There is no system in place to analyze students' strengths and

weaknesses or to offer personalized course and job recommendations based on their abilities.

Overall, the existing system for career guidance is disorganized, lacks intelligent recommendations, and does not support interactive decision-making, making it difficult for students to navigate their career pathways efficiently.

2.2.1 NATURAL SYSTEM STUDIED

The educational and career guidance of the traditional kind guides students using old methods higher studies/career options, mainly face to face counseling, school guidance or the dear ol' fashioned basic online platforms with simple functionalities. While the conventional systems have their advantages faces to face communication with counselors etc., they are so cumbersome and have so many challenges.

1. **Physical Career Counseling:** Students meet with career adviser on the premises of physical counseling centres, wherein the advisers base their advice upon performances and interests in academics or market trends. Although this method enables to offer individualized advice, it slows down the activity and is only limited by number of counselors, and their knowledge in different areas. Issues in physical have things like:
 - a. **Time Consumption:** This takes a lot of time from students because they have to get their schedule for counseling, go to counseling office in campus and then have to spend hours to clear the doubt.
 - b. **Limited Resources:** Career advisors do not necessarily have big and comprehensive databases (of courses) let alone plethora of information on the domain in which companies are hiring.
 - c. **Subjectivity:** The nature of advice varies drastically from one advisor to another since their experience and knowledge might not align fully with the hopes of a student.
2. **Basic Online Career Guidance Systems:** Though a few students began using online platforms for career counseling, these systems generally revisit what on the form itself button without much modification of conventional experience. Most users, In reality users search for course and career related information manually which is really annoying.
 - a. **Limited Personalization:** Basic online platforms also don't cater very well and give you a generic experience in terms of personalized recommendations according to user's preference, academic background or market demand.

- b. **No Integrated Career Planning Tools:** Many young people pursuing pathways through the educational system that are aligned with their career goals tend to be interrupted by information silos leading to a disjointed journey.
- c. **Static Information:** Most platforms display course info, job listings and criteria to apply for something as stale as they come, making it impossible for prospective students to see actual real-time state of jobs.

2.2.2 DESIGNED SYSTEM STUDIED

The Pathway platform is created to deliver on the promise of a personalized, fast and technology enabled user-student experience in relation to their higher studies and career choices from their respective traditional career guidance and educational resource systems.

1. **Personalized Career Recommendations:** The platform uses AI algorithms to give users personalized course and career paths recommendations according to their skills, interests and future ambitions in the field. It allows students to be suggested with relevant and more interesting topics closely that align with their goals, so the decision becomes more meaningful.
2. **Integrated Course and Career Planning:** Pathway includes an integrated planning feature that allows students to explore various educational pathways and corresponding career opportunities. Users can easily visualize the relationships between different courses and the job market, facilitating a more comprehensive understanding of how their choices impact their future.
3. **Dynamic Information Updates:** The system is designed to provide real-time updates on courses, job openings, and entrance requirements. This dynamic information delivery ensures that users have access to the latest data, helping them make informed decisions based on current trends and opportunities in the job market.
4. **User-Friendly Navigation and Interface:** Pathway emphasizes an intuitive user interface that simplifies navigation through courses, career options, and related resources.

2.2.3 DRAWBACKS OF EXISTING SYSTEM

- **Limited Information on Degree Courses:** Existing platforms often lack comprehensive details on degree programs, leaving students with insufficient information to make informed decisions about their higher education options.
- **Insufficient Job Details:** Many current systems provide minimal information about job opportunities, such as roles, responsibilities, and industry requirements, making it challenging for students to understand the career paths available to them.

- **Focus on Crash Courses:** There is often an overemphasis on short-term crash courses, diverting attention from long-term educational pathways that may be more beneficial for students' future careers.
- **Poor Navigation:** Many existing platforms suffer from complicated navigation structures, making it difficult for users to find relevant information quickly and efficiently.
- **Inadequate Course Descriptions:** The descriptions of professional courses are often vague or incomplete, failing to provide students with a clear understanding of what to expect from each program.
- **Lack of Personalization:** Current systems typically do not offer personalized recommendations based on users' academic backgrounds or career aspirations, resulting in a generic user experience.
- **Static Information:** Existing platforms may present static information that does not reflect the most current trends or requirements in education and employment, limiting users' ability to make timely and relevant choices.

2.3 PROPOSED SYSTEM:

The **Pathway** platform is envisioned as a transformative solution designed to enhance the educational experience for students. Unlike existing systems, which often provide limited and generic information, Pathway aims to offer a comprehensive and personalized approach to career planning and educational guidance.

At the core of the proposed system is the integration of advanced technology that delivers detailed course and job information tailored to the unique needs of each user. By utilizing AI-driven algorithms, Pathway will analyze students' profiles, interests, and academic backgrounds, providing personalized recommendations for courses and career paths that align with their individual goals. This targeted approach ensures that students receive guidance that resonates with their aspirations, making the decision-making process more efficient and informed.

In addition to personalized recommendations, the system will incorporate integrated psychometric testing. This feature allows students to gain insights into their strengths, weaknesses, and preferences, enabling them to make more informed choices regarding their educational and career trajectories. By understanding their unique profiles, students can better align their studies with their future career aspirations.

To enhance user experience, Pathway will boast a user-friendly interface that simplifies navigation. Students will be able to effortlessly explore various courses, job opportunities, and educational resources, eliminating the frustration often associated with convoluted platforms. The system will also ensure that all information is dynamically updated, reflecting the latest trends and requirements in education and employment.

Furthermore, Pathway will foster an interactive community where users can communicate and collaborate. Through features like messaging, forums, and feedback options, students and institutions can engage in meaningful discussions, share insights, and support one another in their educational journeys.

A key component of Pathway will be the ability for users to create personal profiles that showcase their achievements, interests, and experiences. This feature not only helps students present themselves effectively but also serves as a valuable tool for institutions seeking to understand their potential candidates better.

To ensure the platform remains robust and user-friendly, the system will include a range of features for administrators and managers. Managers will have the capability to manage course and job details efficiently, while admins will oversee user accounts and maintain the platform's integrity, ensuring a safe and respectful environment for all users.

Overall, the proposed Pathway system is designed to revolutionize the way students approach their education and career planning. By offering personalized guidance, comprehensive resources, and a supportive community, Pathway empowers users to make informed decisions that will shape their futures.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- **User-Friendly Interface:** Simplifies navigation, making it easier for students to explore courses, job opportunities, and educational resources without frustration.
- **Dynamic Updates:** Ensures that all information, including courses and job listings, is current and reflects the latest trends and requirements in education and employment.
- **Comprehensive Resource Availability:** Offers a wide range of information on courses, job opportunities, and entrance exams, helping students make informed decisions.
- **Time-Saving:** Streamlines the process of finding relevant educational and career information, reducing the time students spend on research.
- **Comprehensive Course and Job Information:** The platform provides detailed descriptions

- of courses and job opportunities, helping users make informed decisions.
- **Categorization of Content:** Courses and jobs are categorized effectively, making it easier for users to find relevant information based on their interest.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Objective

The feasibility study aims to analyze the viability, costs, and benefits of developing the Pathway platform, which assists students in planning their careers using React and Node.js. The system is designed to provide a comprehensive understanding of courses and job opportunities to help students make informed decisions.

- **User Roles:**

- **General Users (Students):** The system efficiently handles browsing, viewing course details, know their own strength and weakness through psychometric tests, which help in career recommendation and creating a career route map, and communicating with other users.
- **Managers:** They can manage course and job details, ensuring that the information remains up-to-date and relevant.
- **Admin:** Defined roles for account management and content moderation support smooth operational flow.

- **Ease of Use:**

- The platform focuses on providing an intuitive user interface with easy navigation, addressing common issues in existing systems.
- Emphasis on user-friendly design aims to encourage regular use and enhance user satisfaction.

3.1.1 Economical Feasibility

Cost Estimation:

- **Development Costs**
 - Developer salaries and compensation for project-related roles.
 - Tools and software licenses for development and testing.
 - Payment for quality assurance and testing resources.
- **Operational Costs:**
 - Hosting and server costs for deploying the platform.
 - API usage fees for third-party services.
 - Maintenance and support costs post-launch.
- **Return on Investment:**
 - Through a premium version, the platform can generate revenue

3.1.2 Technical Feasibility

- **Technology Stack:**

1. **Frontend:** React.js
2. **Backend:** Node.js (using Express framework)
3. **Database:** MongoDB

- **Development Tools:**

1. **IDE:** Visual Studio Code
2. **Version Control:** Git

- **Technical Challenges:**

1. Integrating APIs for user communication and data retrieval.
2. Ensuring data security and user authentication through robust security measures.

- **Key Considerations:**

1. **Scalability:** The selected technology stack is highly scalable, accommodating a large number of users, which is essential for an educational platform. Both React and Node.js are recognized for building scalable applications.
2. **Integration:** Seamless integration between React (frontend) and Node.js (backend) is well-documented. MongoDB's flexible schema design supports the dynamic nature of the data (e.g., courses, jobs).
3. **Performance:** React offers fast rendering and dynamic user interfaces, enhancing user experience, while Node.js can efficiently handle concurrent requests, making it suitable for features like messaging

3.1.3 Behavioral Feasibility

Target Audience: The primary users of the Pathway platform are students, educational institutions, and career counselors. Understanding their needs and preferences is essential for ensuring the system's acceptance and effectiveness.

User-Centric Design: The platform should prioritize accessibility, ensuring it is usable for individuals with diverse abilities. This can involve adhering to web accessibility standards and ensuring that all users, regardless of their technical proficiency, can navigate the system effectively

Peer Recommendations: Encouraging satisfied users to share their positive experiences can enhance the platform's credibility and attract new users. Implementing referral programs or incentives for users who successfully refer others can be an effective strategy.

3.1.4 Feasibility Study Questionnaire

- **Is today's student facing in choosing career? Why?**

Yes, The main reason for this is the too many courses are there in the market. Each year the number of courses and job are created and more specifications in each courses and jobs are coming.

- **Is generation gap made any difference in choosing career?**

Yes, 10years ago student choose their career based on their financial stability and parents also influenced, but today the situation is not the same. Peer group influences them more in such decision.

- **Today, how does a student take a career options, on what basis?**

Today, students are influenced by their peer groups. Most of the take the course what their friends or peers taken in order to fit in.

- **What is the primary objective?**

The main objective is to help students and professionals make informed decisions about their education and career paths by providing comprehensive information on courses, job opportunities, and career-related guidance.

- **What makes Pathway different from traditional career guidance methods?**

Pathway is a centralized online platform that offers access to detailed course and career information, making it easy to research, compare, and choose educational and career options, all in one place.

- **How does Pathway support career planning for users?**

Pathway provides users with information on courses, career opportunities, and educational pathways, helping them make decisions aligned with their interests and career goals.

- **What types of activities can users perform on Pathway?**

Users can search for courses, view career options, post articles, provide feedback, and interact with the platform by liking, sharing, and commenting on posts. Also an AI and ML based systems help in achieving careers

- **How does Pathway ensure that course and job information is up-to-date?**

Managers and admins can update course and job information regularly. This ensures that users always have access to the latest details.

- **What kind of support does Pathway offer for making career decisions?**

Pathway provides structured career information, allowing users to explore and understand various educational and career paths and make well-informed choices.

- **How can students get to know about the career in deep?**

By talking to the persons, who are now at their desired career help them know more about career.

3.1.5 Requirement Gathering

Project Overview

The Career Pathway project is an educational web platform aimed at assisting students in making informed decisions about their higher education and career paths. The platform provides comprehensive information on courses, job opportunities, and mock tests, including psychometric tests. It enables users to search for relevant courses, explore career options, and interact through blogs and feedback systems. The system incorporates role-based access control for different users, ensuring streamlined operations and efficient data management.

Main Objectives

1. Implement a secure login system with role-based access for Admin, Manager, and General Users (students and institutions).
2. Allow Admins and Managers to add, update, and manage courses and job listings.
3. Enable users to explore and filter courses and career options based on categories and subcategories.
4. Provide an interactive blog system where users can post educational articles, like, share, and comment.
5. Facilitate psychometric tests to help students determine suitable career paths.
6. Implement a reporting mechanism for inappropriate content on blogs.
7. Ensure a centralized data management system for efficient tracking of user interactions.
8. Develop an admin dashboard to monitor platform usage, manage reported content, and oversee user activities.
9. Implement a feedback system allowing students to review courses and career paths.
10. Enable institutions and users to send messages for direct inquiries.

Target Audience

- Students seeking course and career guidance.
- Educational Institutions providing course information.
- Career Counselors and Admins managing content and psychometric tests.
- Platform Admins overseeing system functionalities.

Modules

1. Admin

- Manage access for users, including Managers and General Users.
- Approve or remove courses, job details, and blog posts.
- Handle reports and block/unblock content.
- Oversee psychometric test modules and results.
- Monitor platform usage and manage overall system settings.

2. Manager

- Add and manage course and career details.
- Modify or remove job postings.
- Moderate user feedback and respond to queries.

3. General Users (Students & Institutions)

- Register and log in to explore career and course options.
- Filter and search courses based on categories and subcategories.
- Participate in psychometric tests to receive career recommendations.
- Post and engage with blogs (like, share, and comment).
- Report inappropriate blog content.
- Send messages to institutions for inquiries.

4. Psychometric Test Module

- Provide structured tests to analyze students' strengths and interests.
- Generate career recommendations based on test results.
- Store and allow access to past test results for future reference.

5. Reporting and Feedback System

- Users can report inappropriate blogs or comments.
- Admins can review, block, or remove reported content.
- Students can provide feedback on courses and career paths.

6. Messaging System

- Enables students to directly contact institutions for course-related inquiries.
- Facilitates communication between users regarding career guidance.

System Ownership The system is managed by an educational organization responsible for maintaining course and career information. Institutions contribute by adding details about courses, while students interact with the platform for guidance.

Data Collection Contacts

1. Name: Rajendran

Role: Teacher, Career counsellor

Questionnaire for Data Collection

1. What are the primary responsibilities of the admin in the Career Pathway platform?
 - Managing user access, approving/rejecting courses and jobs, handling reported content, and overseeing system security.
2. What challenges might the admin face in maintaining the system?
 - Ensuring data accuracy, handling user feedback and reports, and preventing misuse of the platform.
3. What functionalities are available to students ?
 - Searching for courses and careers, taking psychometric tests, posting and interacting with blogs, messaging institutions, and providing feedback on courses.
4. How should the psychometric test module function?
 - It should present structured questions, evaluate responses, and generate career recommendations based on the results.
5. What measures should be in place to manage reported content?
 - Admins should review reports, take necessary action (block/unblock content), and maintain a log of reported incidents for reference.
6. How should the feedback system be managed?
 - Feedback should be stored in a database, categorized by course/job, and visible to other users for informed decision-making.

3.1.6 Geotagged Photograph

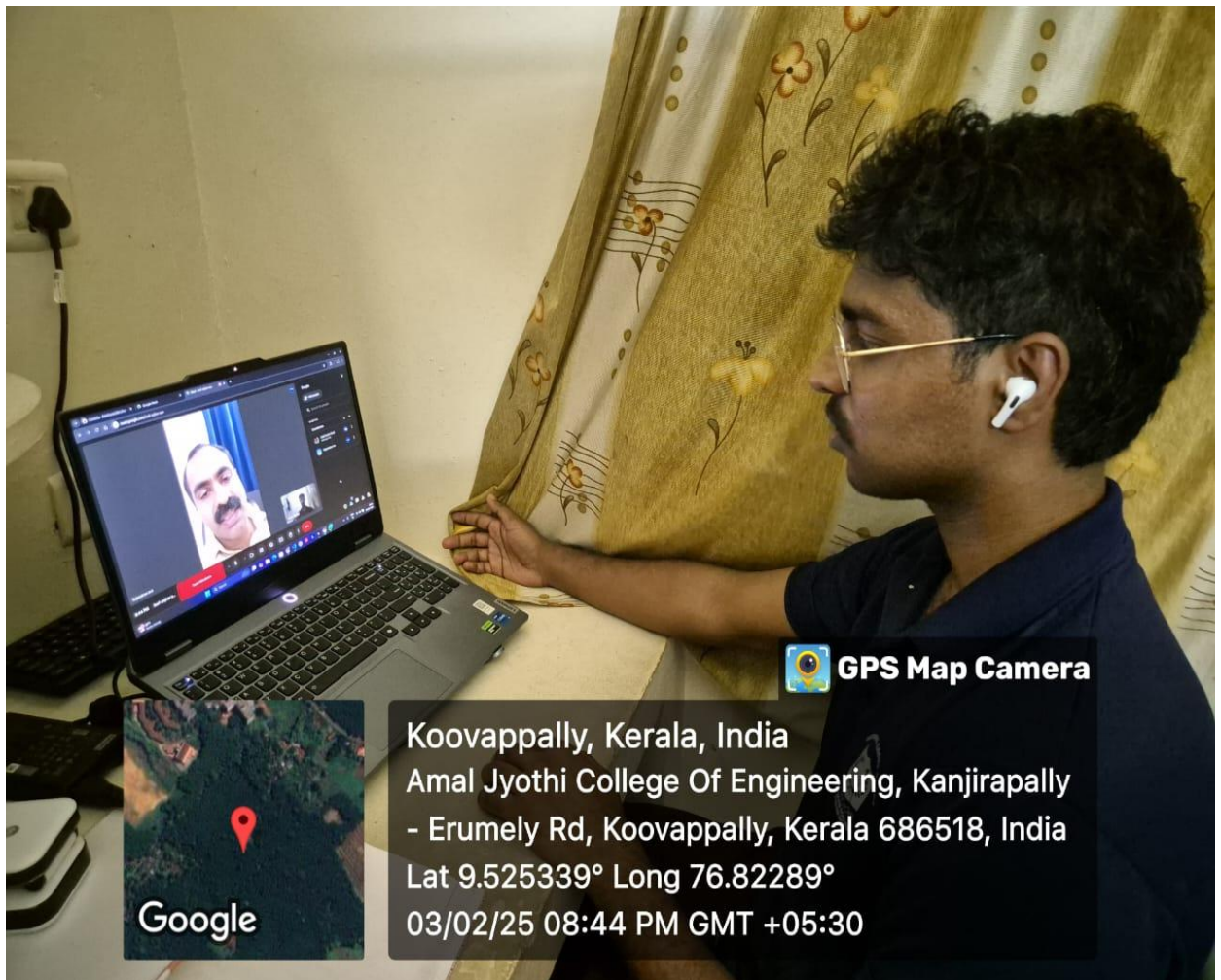


Fig 3.1.5.1

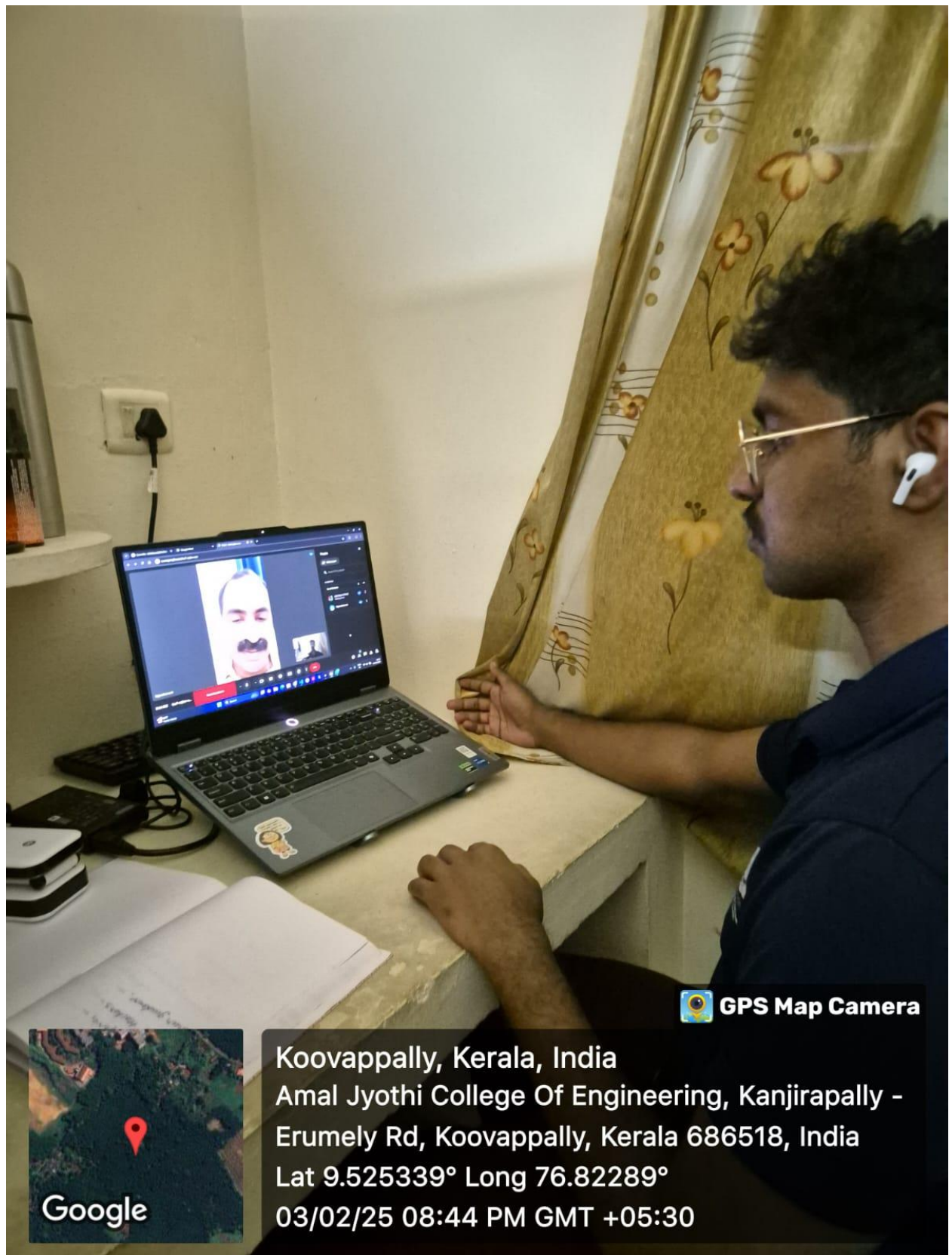


Fig 3.1.5.2

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel® Corei3

RAM - 4 . 0 G B

Hard disk - 2 5 6 G B

3.2.2 Software Specification

Front End - REACT JSX

Back End - NODE JS

Database - MONGO DB

Client on PC - Windows 7 and above.

Technologies used - JS, JSX, JSON, CSS, Restful APIs
API: RazorPay, FireBase, Gemini API
ML Technologies: Random Forest

3.3 SOFTWARE DESCRIPTION

3.3.1 React js

React js(usually called as React) is a high level JavaScript library/ framework for building user interfaces, hits really hard on efficiency and experience while working with single page application. React is an open-source JavaScript library created and run by Facebook for building complex and large web applications which can update views in response to data changes without actually reloading page. Is component based, developers can take control of complex UIs by deconstructing them into small components with a single state and/or logic.

React, one of the greatest features- virtual DOM(Document Object Model) in react is to optimize performance on the backend by re-rendering only a part of the page that changed. React does not re-render the whole DOM every time there is any change, instead it keeps a virtual DOM in memory. Data changes, React compares the new virtual DOM with earlier one and updates only those elements which are updated in this case making it faster and scalable.

React also has a strong emphasis on declarative programming, which helps developers to figure out how the app should behave. For a declarative language, developers specify what should be the

UI for a given state and React takes care of rendering. This results in cleaner, more readable code, especially for larger applications.

React is also very polymorphish in, it plays nice with other libraries or frameworks (Redux for example or state management, Next.js for server-side rendering etc.) The ecosystem is vast — communities contribute tons of libraries, tools and community resources which makes building dynamic applications on web much more easier for developers. The component-based architecture of React, combined with its fast performance and unlimited scope of application is why this option remains one of the most sought after in front-end development nowadays.

3.3.2Eg. Mongo DB

MongoDB is a powerful NoSQL database that is frequently used for flexibility, high scalability and ease of use where multitude of applications need to deal with large data volumes, both structured & unstructured/semi-structured data for example.

While traditional relational databases organize data in tabular rows and columns, MongoDB stores flexible data in documents called BSON (Binary JSON) that lends itself to data, and therefore is one of the best databases for different kinds of data formats. The document-oriented nature of this allows for a neater data modeling, especially where data type structure might evolve (as you would in say modern web apps).

Perhaps the greatest strength of MongoDB lies its horizontal scalability, which fits it for distributed applications with unbounded growth in data. This makes distribution and sharding (scalable data distribution) across multiple servers in Mongo's architecture so you get high availability, performance even with exponential growth of database. This is a great scalability (in case of cloud based and distributed systems and the data reside across different geographies as well servers iteration.

MongoDB has also got great querying utilities. Data, despite its being non-relational can be filtered and aggregated on complex operations even further through queries as well as filtering, aggregations being easily performed by developers. The indexing capabilities of the database enhance query performance and it provides for all kinds of operations the full power of text search, geospatial queries and real-time analytic support. Also, it comes with replication and high availability support through the replica set feature, which guarantees that the data is always backed up and available in case of hardware fault.

3.3.3 Express and Node js

Node.js is a robust open-source JavaScript runtime where a developer can execute the javascript outside of the browser. Node.js is built around Google Chrome's V8 JavaScript engine

that aims to deliver quick and efficient server-side applications. It is an event-driven, non-blocking I/O model enables to handle multiple requests at once, real time applications or any application having heavy I/O oriented operations like APIs or chat application.

A principal feature of Node.js is its Asynchronous, Single-Threaded Architecture. Node.js, contrary to the multi-threaded model of normal server-side environment that handle concurrent request in a single server one could serve all requests using the single thread with Node. It allows Node to easily chew through thousands of requests simultaneously, making it great for highly scalable and fast applications. Another huge advantage is the package ecosystem of Node.js that is maintained by npm (Node Package Manager) providing an extensive atlas of open-source libraries and modules where developers can import very easily.

Express is a minimalist web application framework built on Node.js designed to help people get started with web server/ APIs in a fast way (basically if you're just building that). Server-side JavaScript applications are built on top of Node.js, Express.js is a very thin framework that takes away most of the boilerplate involved in the HTTP request handling, routing and middleware registration from you. Express is easy to use, flexible and fast framework for building RESTful APIs as well as regular web applications.

Routing is one of the main parts of Express.js, this library enables developers to define URL routes and the handler functions being called when those URL routes are matched. Thus, it is very convenient to design such code that is clean, organized and tractable for different HTTP methods (GET, POST, PUT, DELETE) and paths. In addition, Express also provides middleware (functions that execute during the request-response cycle) that allow the developer to keep their code modular (like authentication, logging request process, parsing requests and error handling). Node+Express.js is a synergy that can help to create top-notch full-stack JavaScript based applications with the Node as classic server and Express simplifying web servers & APIs.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

System design phase marks the transition from conceptual planning to the structured creation of a working solution, laying down the foundation for successful implementation. Design in software engineering refers to the process of defining a system in sufficient detail to facilitate its construction and ensure it meets desired requirements. This phase serves as the blueprint for developers, focusing on how the system functions, interacts, and ultimately fulfills its purpose.

System design is essential for translating user needs and functional specifications into a structured form that guides the coding, testing, and deployment stages. By leveraging various techniques and principles, system design allows for a comprehensive depiction of how different components interact within the system and how each piece of functionality aligns with the project's goals.

In software projects, like the Pathway platform, the design phase provides clarity and structure, moving the focus from user needs and requirements to the practical considerations necessary for development. It is typically divided into two main parts:

1. Logical Design – Defines the system's structure, processes, and data flow without considering technical constraints, focusing on what the system must do.
2. Physical Design – Converts the logical framework into detailed specifications, including hardware, software, and infrastructure requirements, providing the technical blueprint for the system's realization.

A well-crafted design phase is essential to ensuring efficient, accurate, and high-performing software, setting the stage for a successful system build.

4.2 UML DIAGRAM

UML (Unified Modeling Language) is a well standardized language for specifying, visualizing, constructing, and documenting the elements of software systems. UML [1], the Universal Modeling Language, was created by Object Management Group (OMG) and the first draft of UML 1.0 specification was submitted to OMG in January 1997. UML is not a programming language (just as C++, Java or COBOL, not what you program with) compared to regular programming languages. Rather, it is a visual language to use in defining Software design sketches.

The UML is generally a versatile and systematic visual modeling language for the purpose of systematizing the visualization, specification construction and documentation of software systems. While its primary use is in modeling software systems, UML's applications are not limited to this

domain. It can also model and reason about processes in a wide range of situations including non-software such as manufacturing unit processes for.

Although UML is not a programming language, it can be used with tools that generate code in several other programming languages from the UML diagrams. The UML consists of nine core diagrams to represent different facets of a system.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case is a tool you use to make sense out requirements in to something more manageable for example product delivery website or creating one. The above tools are visualized using Unified Modelling Language (UML) Represented by use case diagrams for an established way to model real-world things and solutions.

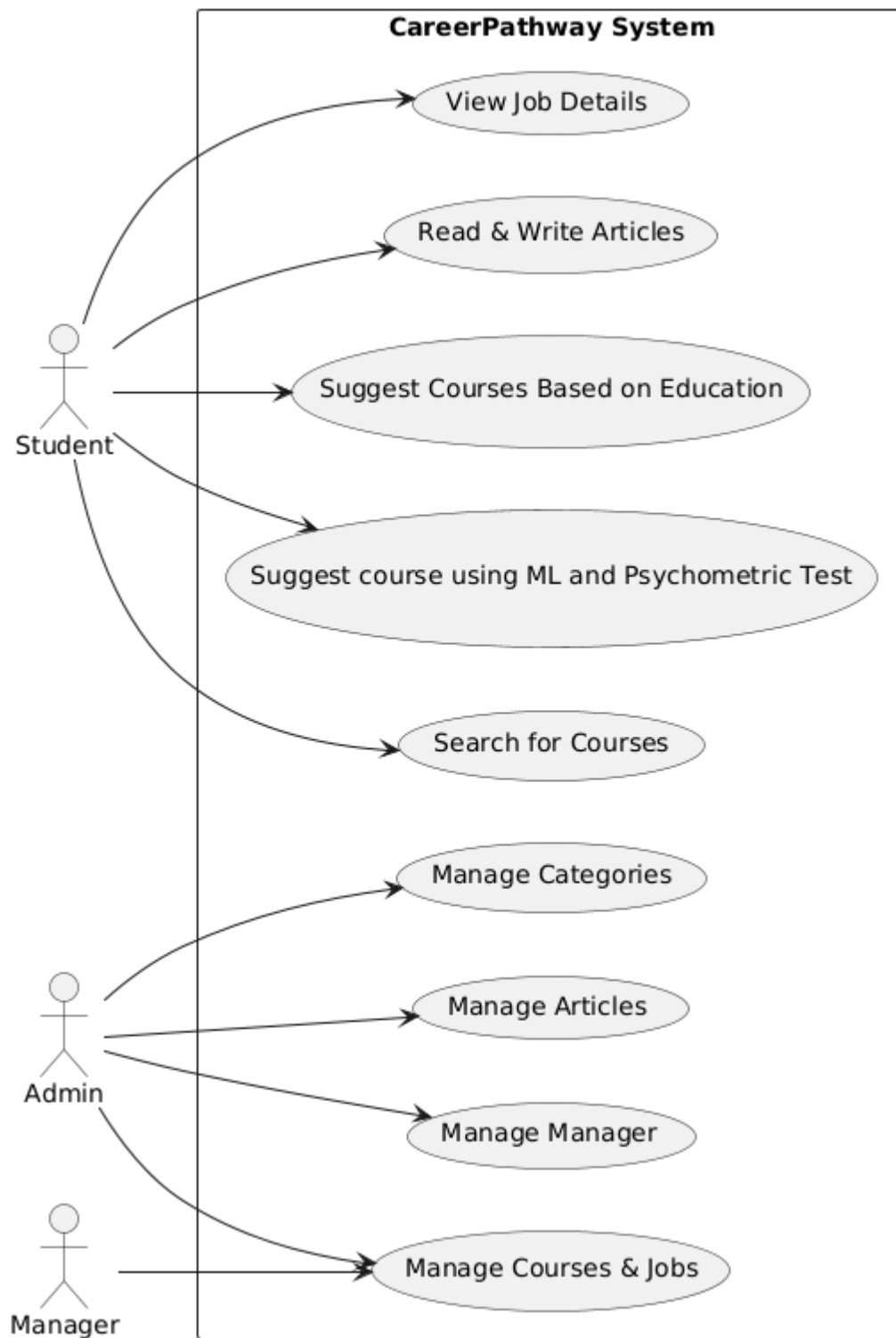
A Use Case Diagram has these major elements:

- The boundary that separates the system from its environment.
- Actors — human actors playing different roles in the system.
- Inter actions of different parties or factors in certain situations or problems.
- The main goal of use case diagrams is to clarify functional specifications of a system.

Few Tips that Can Be Followed While Making Use Case Diagram as Effective:

- To give clear, understandable names to Use Cases and Actors.
- Ensuring Well Defined Relationships and Dependencies.
- Include only the necessary relationships to keep the diagram sane.
- Use explanatory notes when needed for the important details

HL: use explanatory notes to explain details required.

*Fig 4.2.1*

4.2.2 SEQUENCE DIAGRAM

A sequence diagram depicts the order in which objects talk to each other (which shows a progression of events from top to bottom). This diagram is also called an event diagram or event scenarios. Sequence diagrams aims to clarify with the interrelationship of many parts in the system, and their precise sequence of execution. Business, as well software professionals use these diagrams to both scratch out requirements of new and existing systems in explanation and visual form.

Sequence Diagram Notations:

- i. Actors — Actors in UML diagram, are people who use a system, and its parts. The actors are not (represented in a UML diagram, as they are exterior to the modeled system). Also they are peeps whose story it will be since they are role-players in the story, i.e people and external entities. In UML diagram the actors are represented by whatever simple stick figures. You can have multiple people in a diagram showing the flow of events but as one goes to one specific incident.
- ii. Lifelines — lifeline to each element of a sequence diagram is represented on diagram by a lifeline with the lifeline components in the top of figure.
- iii. Messages – Messages exchanged between objects are organized on the lifeline as messages take place one after another in sequence. Messages are represented by Arrows that form the basis structure of a Sequence diagram.
- iv. Guards: Guards (also within the UML) are used to represent different types of conditions On them messages may be restricted if certain conditions hold and give you hints about the rules which are in a system / process.

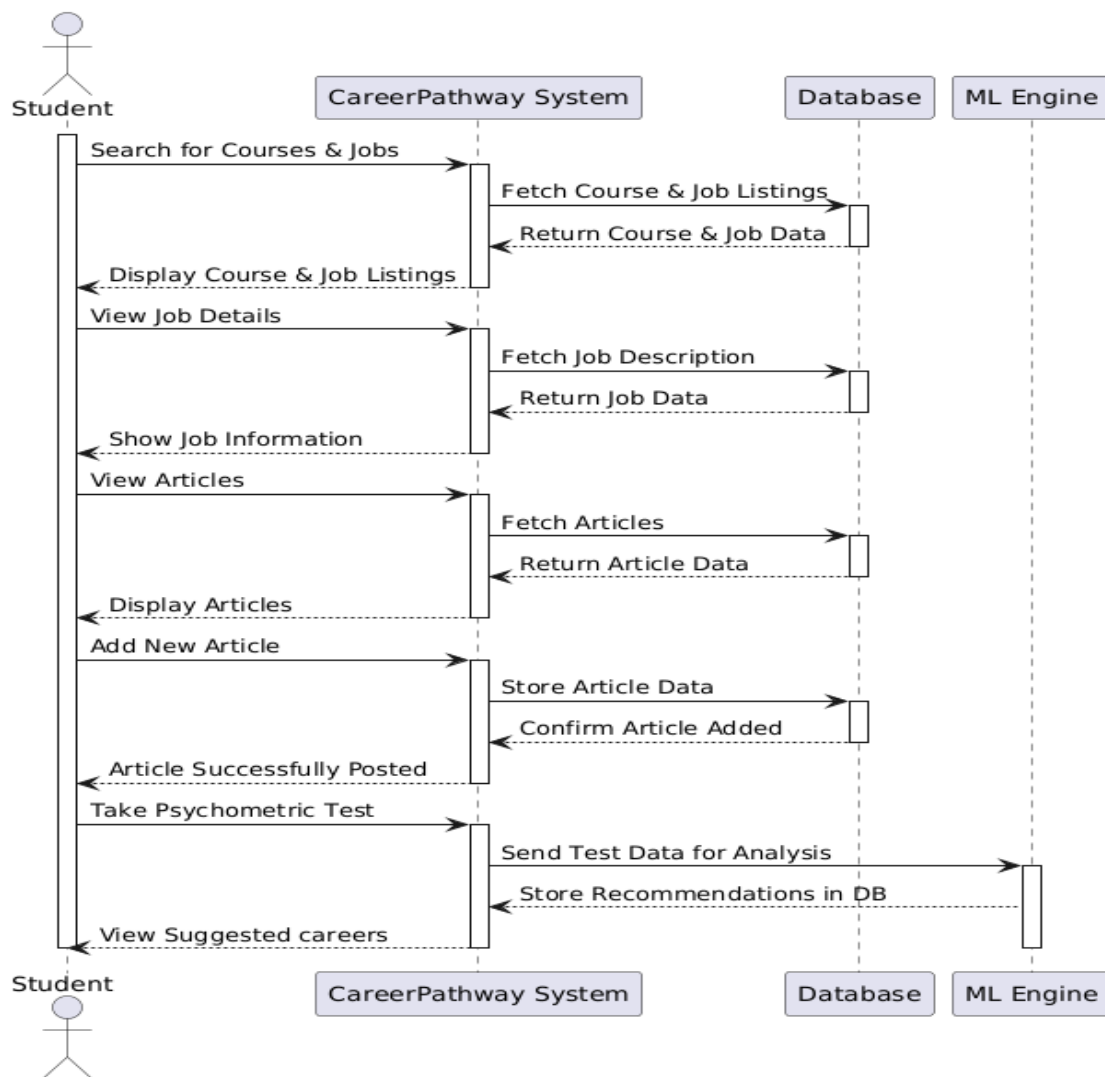


Fig 4.2.2

4.2.2 State Chart Diagram

A State Machine diagram (also called state chart) being a pictorial representation of the states an object can be into within a system and the path through these states. It is a trace of the behavior of the system, showing the shared behaviors of a group of entities (be it team, students in one class, group audience or even the whole organization). State machine diagrams are a helpful means for illustrating interactions of different participating components within a system detailing how objects change in response to events and make clear different states in which each and every entity or component can be.

A state machine diagram covers the following Notations within:

- Initial State : Black Circle initially: Initial state-demonstrated through black circle as an umbrella state for process.
- Result state: This is shown with a filled-in circle within a circle to represent the end of a process.
- Decision box: A diamond shaped element that provides a decision to make based on guard's

evaluation.

- Transition: Transition is when the switch in authority/state due an event Transitions are shown as labeled circling arrows that indicate the event triggering the transition.
- State box: Represents the element within a group of that state at a particular instant. They are usually rectangles with rounded corners such as.

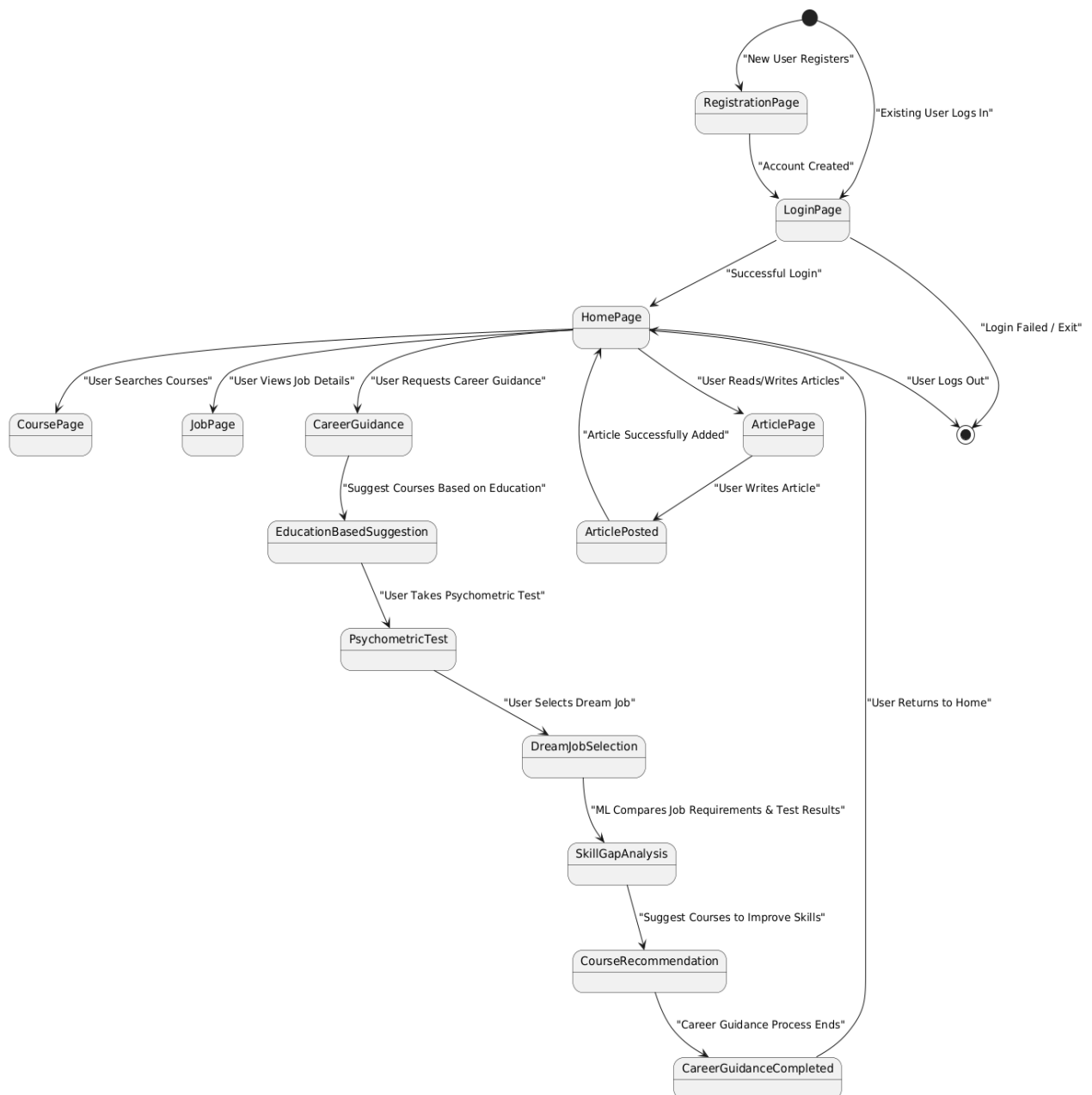


Fig 4.2.3

4.2.4 Activity Diagram

An activity diagram illustrates the shared and sequential progression of events in system lifecycles. Helps to know the activities happening, making it clear one task going to another activity Task sequences is the main theme of activity diagrams and it can model different kinds of flows such as

sequential, parallel and alternative paths for this. To support these flows in activity diagrams, forks and join nodes are used which also complements the view to show workload of a system that the system is designed to illustrate more precisely.

Activity Diagram Key Components are:

- a) Activities: Activities aggregate behaviors into one or more actions, so we have the network of different with steps interconnected. Actions that connect these steps are lines, the occurrences at that point are all the step-by-step breaks. Actions includes tasks, driving factors and process resources.
- b) Activity Partition/Swim Lane :In the swim lane of an activity diagram similar tasks are categorized into rows or columns, but can only be arranged into columns not mandatory for every activity diaram. They can oriented in vertical or horizontal, although they are not all require in every activity diagram from mandatory.
- c) Forks: Fork nodes allows to run of various branches of a partical task. They are the place, where one input turns into several outputs which is analogous to different factors in influencing a decision.
- d) Join Nodes → Another kind of fork node which uses a Logical AND where all incoming data will converge to one place for synchronization is different from fork node. ..

Activity Diagram Notations are as follows

- Initial State — the starting or 1st step in a process
- Final State: Denotes the end of actions none further progress.
- Decision Box: To assure that our activities are routed correctly.
- An action: Task or activities to be executed on in the process.

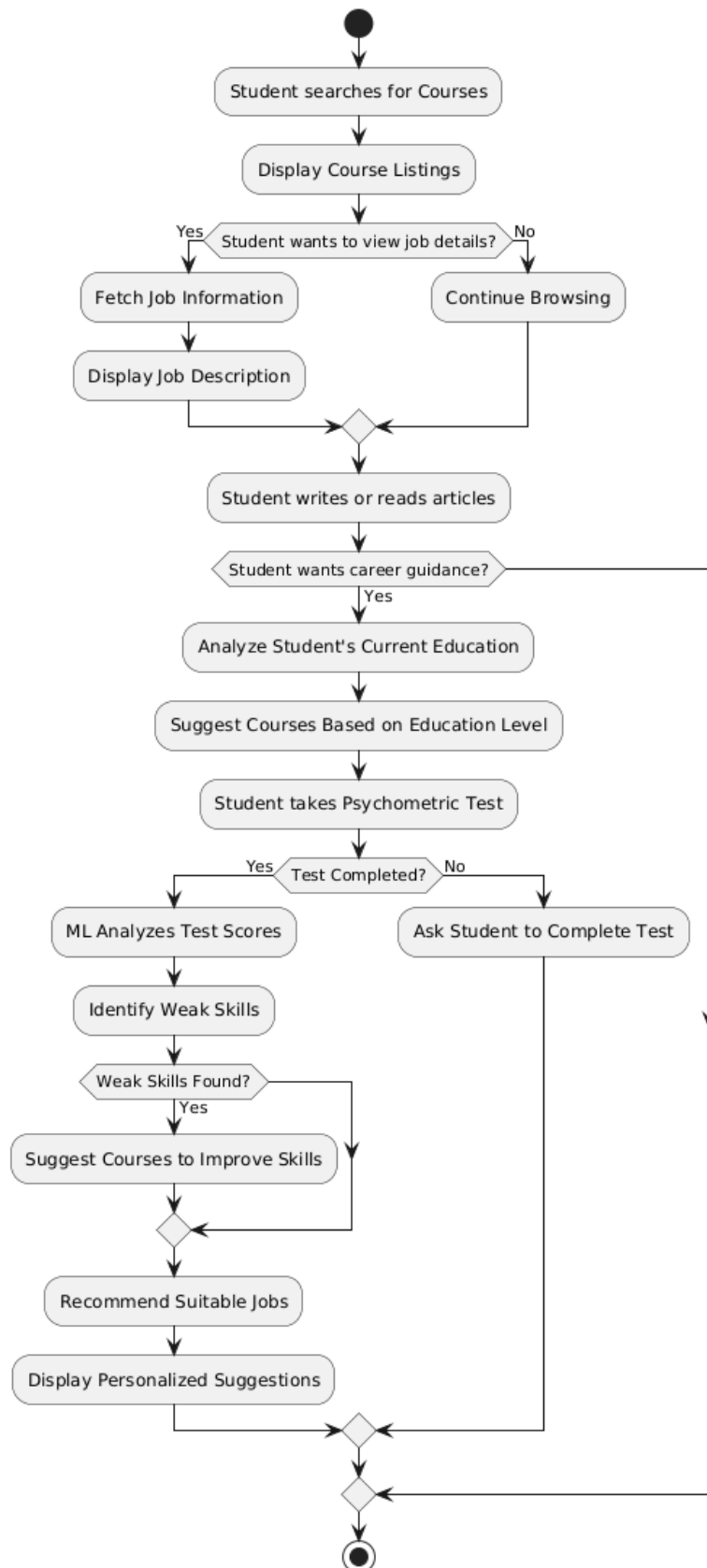


Fig 4.2.5

4.2.5 Class Diagram

A class diagram is meant as a blueprint for the Application at Static State, depicting what components it will have and their relationship when system is actually down. It dives into the system design, displaying the different pieces the system is made from and their relationship. A class diagram, essentially is a recipe for developers as they build functional applications in software.

Important Elements of Class Diagram:

- **System Overview:** A class diagram is a high level representation of software system further defines its compile pieces, relationship and process collaborations. A good organizational structure for names, attributes and methods in software development.
- **Structural visualization:** The class diagram provides the structural view of system architecture which combines classes, associations and constraints.

Components of a Class Diagram:

A class diagram consisted partly of three major areas:

- **Upper Section:** Class name, a collection of objects that have similar properties, attributes and behavior. Capitalize initial letter of class name and place it at centre show abstract class title in slanted writing style are some guidelines to demonstrate groups of objects.
- **Middle Section:** This portion details the attributes of the class and their visibilities denoted public (+), private (-), protected (#), ~ as package.
- **Lower Section:** Lower segment goes into the methods or operations of a class, with each method in the list starting on a new line Of Class demonstrate its relationship with data.

In UML, relationships within a class diagram fall into three categories:

- **Dependency:** Signifying dependence of change in one element on change in another.
- **Generalization :**Representing a hierarchical relationship One class as general and another as specific.
- **Association:** Form evident connections between elements.
- **Multiplicity:** One is the default value for the number of instances which may have certain characteristics develop a la list.
- **Aggregation:** An aggregation is a collection with a relationship available as association.
- **Composition:** Composition is a sort of aggregation, sort of It tells the parent needs

its child to exist or rather one cannot work without the other.

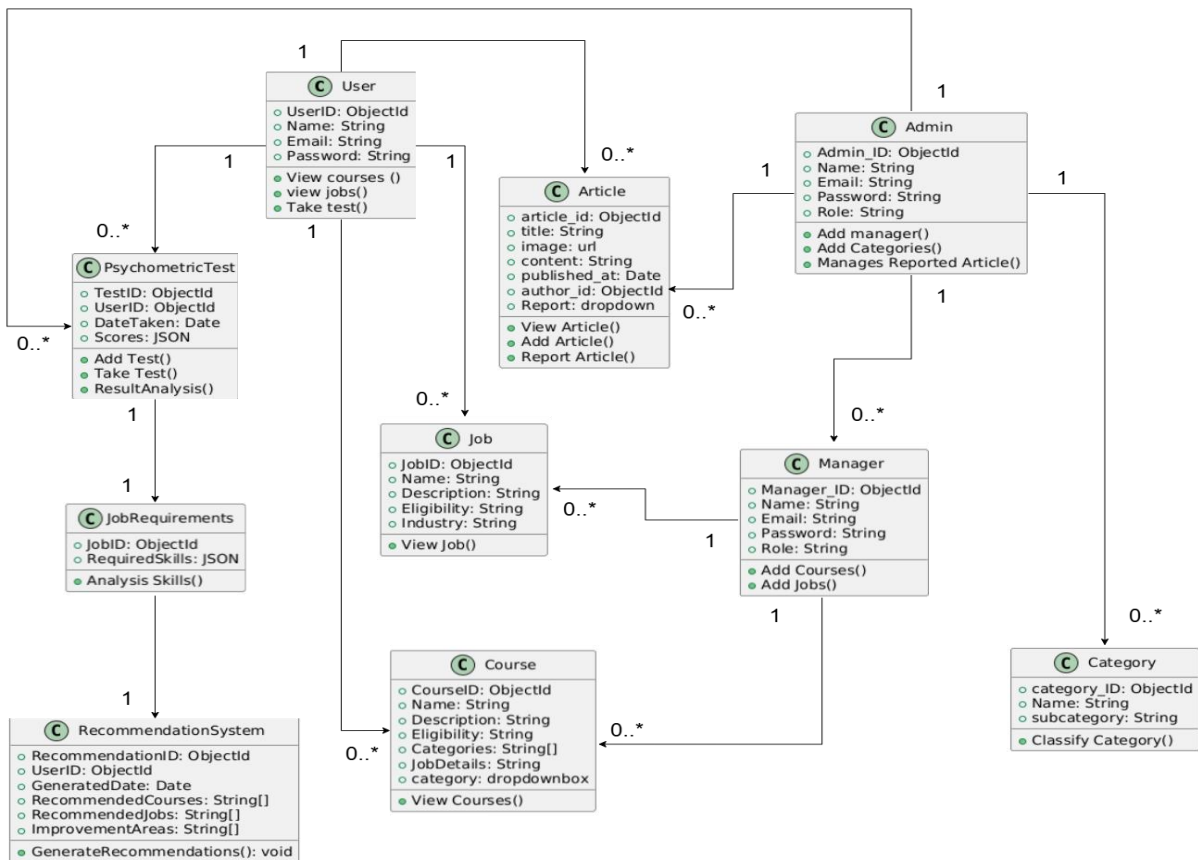


Fig 4.2.5

4.2.6 Object Diagram

Object diagrams are generated from class diagrams and depend on them for their visual. They give a picture of a set of objects in association to some class. An object diagram is simply a point in time snapshot of a object-oriented system. The differences and similarities between object diagrams and class diagrams While true, these have distinctions as well. Whereas a Class diagram is really pretty high level where it shows the classes as is. Class diagrams provide an easy to understand abstraction of system functionality and architecture that the workings of individual objects are hidden.

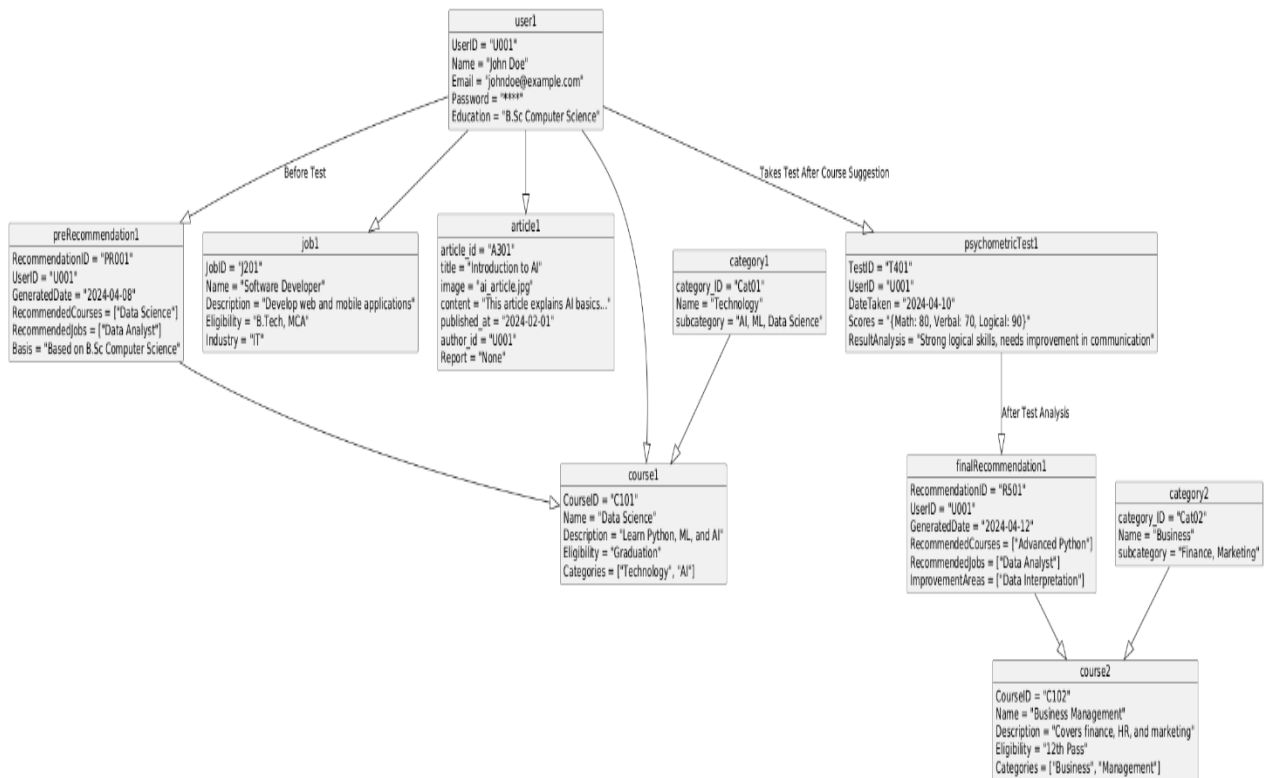


Fig 4.2.6

4.2.7 Component Diagram

A component diagram is a tool to componentize large objects of a complex system. It provides a visual depiction of the system, in terms of internal to nodes systems like programs, documents and tools called the system. The associations and organization of the elements of system diagram gives working system Graphic.

Within a component diagram, the component is an interchangeable system unit part that can be varied and can function on its own. It keeps the curtain drawn on its inner workings and needs a special way to do its job, much like a magic concealed box that really only works properly when operated.

Related Notations for a Component Diagram:

- A component
- A node

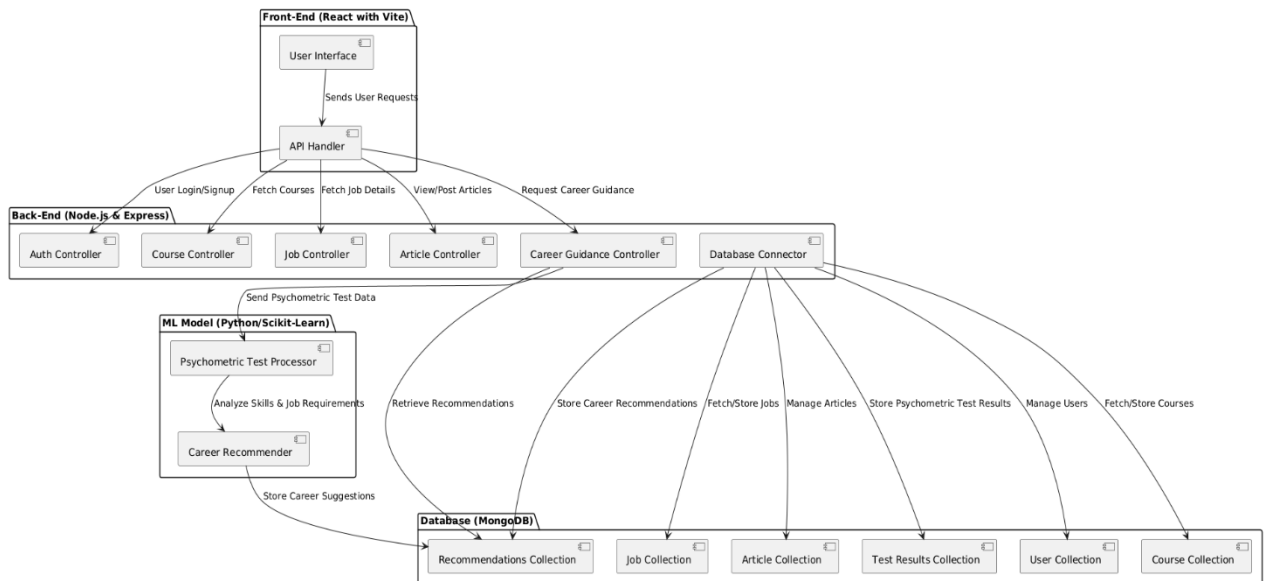


Fig 4.2.7

4.2.8 Collaboration Diagram

A collaboration diagram is a collaboration tool to depict the relationships between objects to a system. It is another way of saying the same thing than a sequence diagram, though. A collaboration diagram creates a snapshot of the object structure within system, rather than show the flow of messages between objects. These are based on the notion of object-oriented programming where object have attributes and are related to each other. Collaboration diagrams more or less represent the object architecture of system in a visual format.

Key elements in a collaboration diagram include:

- **Objects:** An object is devoted to their (s) names and class(s) (under-one), separated by a colon. Object: in a collaboration diagrams the objects are instances of a class; they will specify both the name and class of that instance. Not all classes are going to need an object representation and in fact, a class can have many different instances.
- **Actors:** These are pretty important in collaboration diagrams as they are the ones that initiate interactions.
- **Links:** Links are associations between object and actors, in the form of an instance It shows the relationships between the objects through which messages are passed between objects. An object navigates through links (solid lines) to reach other objects.
- **Messages:** Objects communicate via messages (object-to-object communication including an information carried in it and have a sequence number to uniquely identify them.) Such messages are then shown as labeled arrows next to the corresponding links, from sender to receiver. The messages must be directed and the receiver must know that specific message.

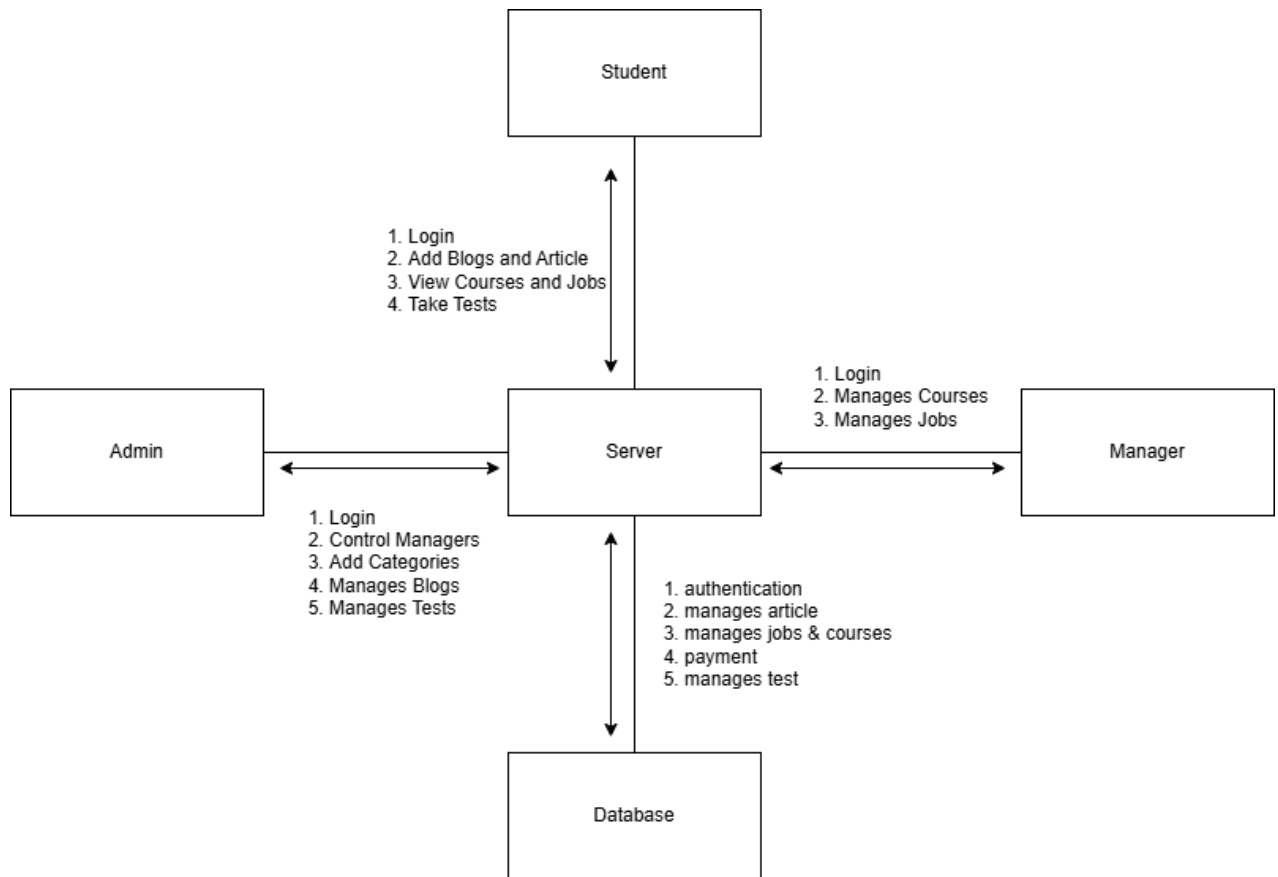


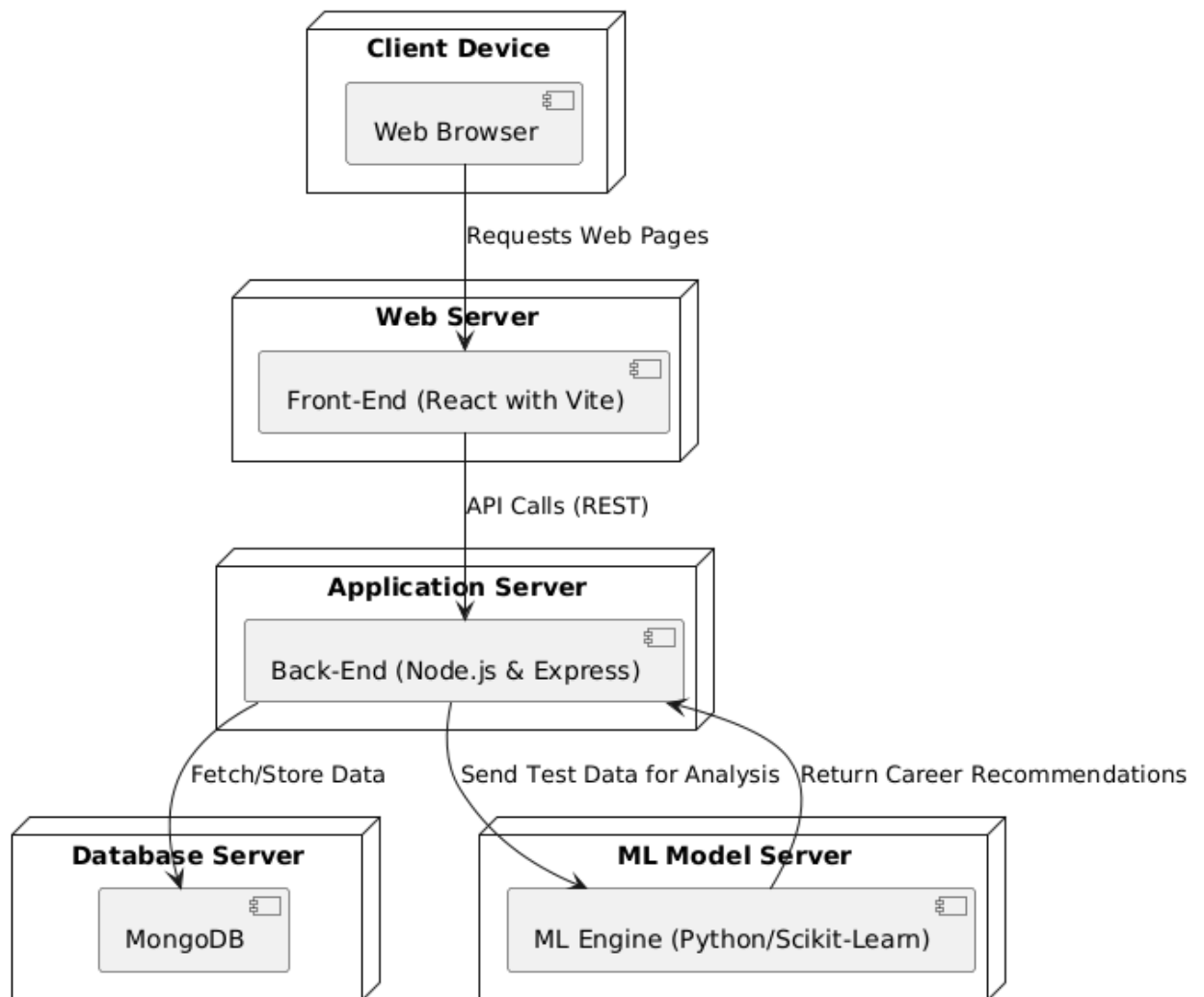
Fig 4.2.8

4.2.9 Deployment Diagram

The deployment diagram shows where software goes in the physical machines or servers as a diagram. It provides the static image of the system in the nodes and its connectivity. This diagram delves into the mechanics of programs being put on computers, offering what the software is built this way to sync up with physical machine system.

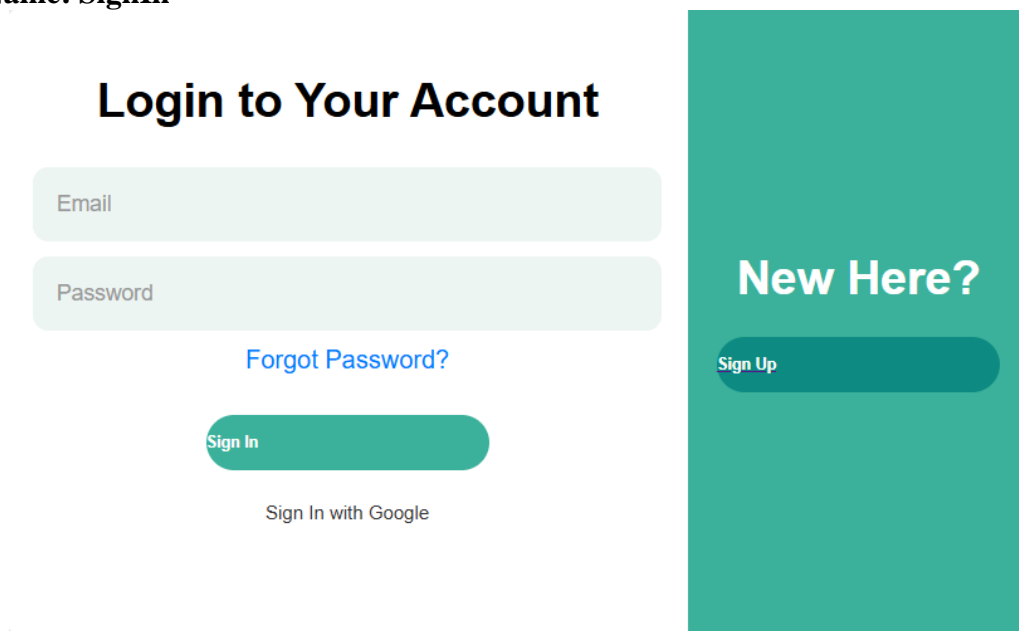
Deployments Diagram Notations are the likes:

- A component
- An artifact
- An interface
- A node

*Fig 4.2.9*

4.3 USER INTERFACE DESIGN USING FIGMA

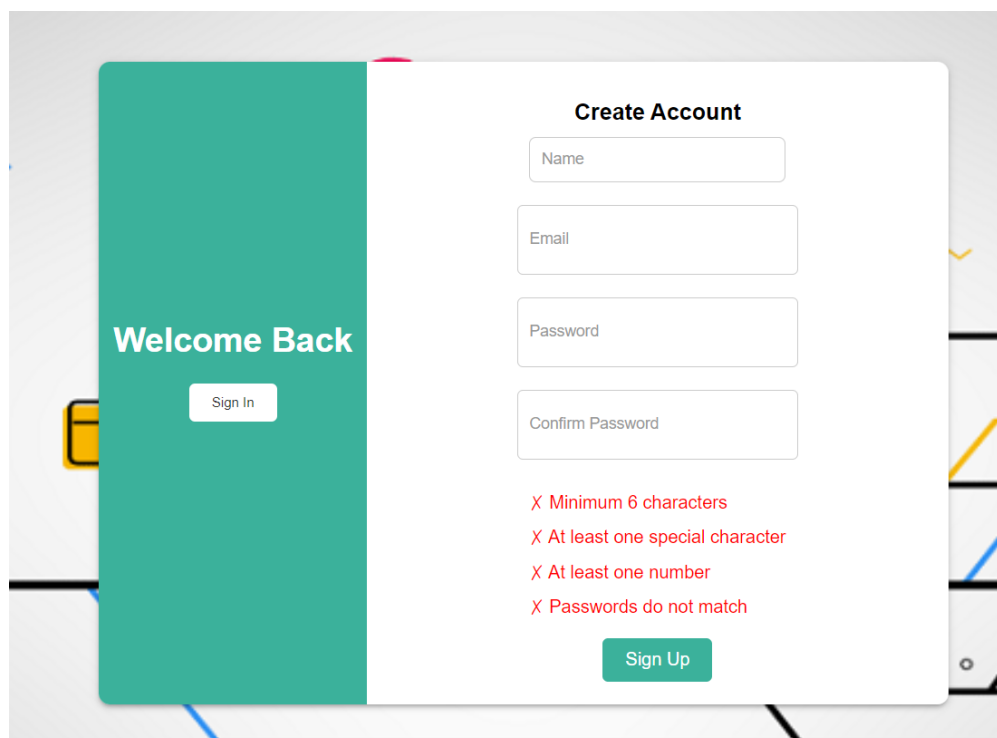
Form Name: SignIn



The image shows a login form titled "Login to Your Account". It features two input fields: "Email" and "Password". Below the "Password" field is a link "Forgot Password?". A green "Sign In" button is positioned below the "Forgot Password?" link. Below the "Sign In" button is a text link "Sign In with Google". To the right of the login form is a vertical teal banner with the text "New Here?" and a "Sign Up" button.

Fig 4.3.1

Form Name: SignUp



The image shows a sign-up form titled "Create Account". It features four input fields: "Name", "Email", "Password", and "Confirm Password". Below the "Confirm Password" field are four error messages in red text: "X Minimum 6 characters", "X At least one special character", "X At least one number", and "X Passwords do not match". A green "Sign Up" button is positioned below the error messages. To the left of the sign-up form is a vertical teal banner with the text "Welcome Back" and a "Sign In" button.

Fig 4.3.2

Form Name: Mock Tests

The screenshot shows a web application interface for mock tests. On the left is a vertical sidebar with a logo at the top and five menu items: Home, Jobs, Courses, Blogs, and Tests. The main content area is titled 'AVAILABLE TESTS' and contains three cards. Each card represents a different test category: MATHS, LOGICAL REASONING, and VERBAL ABILITY. Each card displays the total marks and duration, and a 'Start Test' button.

Test Category	Total Marks	Duration	Action
MATHS	15	15 mins	Start Test
LOGICAL REASONING	5	5 mins	Start Test
VERBAL ABILITY	5	5 mins	Start Test

*Fig 4.2.3***Form Name: Mock Test : Maths**

The screenshot shows the 'MATHS' section of the mock test interface. The sidebar is identical to the previous figure. The main content area is titled 'MATHS' and includes the duration (15 minutes) and total marks (15). A description states: 'Contain Problems On Train, Time and Distance, Time and Work'. There are two questions, Q1 and Q2, each with two radio button options and a 'Marks: 1' label.

MATHS
Duration: 15 minutes
Total Marks: 15
Description: Contain Problems On Train, Time and Distance, Time and Work

Q1. A train running at the speed of 60 km/hr crosses a pole in 9 seconds. What is the length of the train?

☐ 120 metres
☐ 150 metres
Marks: 1

Q2. A train 125 m long passes a man, running at 5 km/hr in the same direction in which the train is going, in 10 seconds. The speed of the train is:

☐ 50 km/hr
☐ 55 km/hr
Marks: 1

Fig 4.2.4

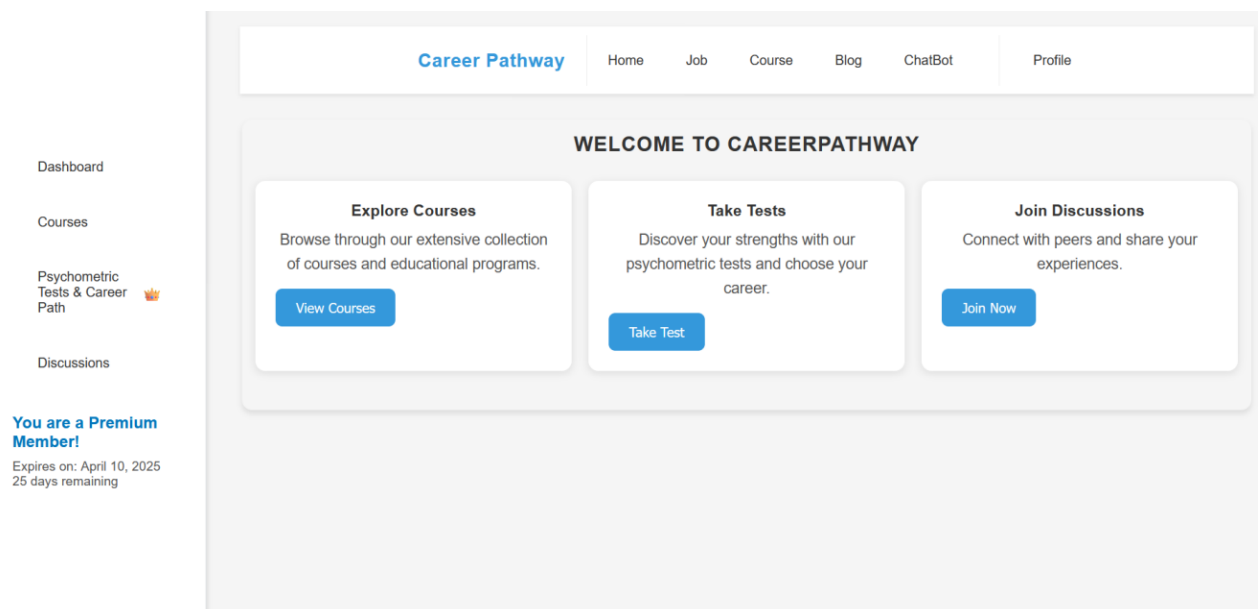
Form Name: Job List

The screenshot displays the 'Job List' page of the Career Pathway application. On the left, a sidebar contains the 'Career Pathway Logo' and a 'Filter by Category' section with checkboxes for Undergraduate (BA, BSc), Post Graduation (M.Sc, M A, MD, Professional), and a 'View Details' button. The main content area features a navigation bar with 'Career Pathway' (active), Home, Job, Course, Blog, ChatBot, and Profile. Below the navigation bar is a section titled 'AVAILABLE JOB OPPORTUNITIES' containing three job cards: 'Data Scientist' (Undergraduate, Location N/A), 'Software development' (Post Graduation, Location N/A), and 'Marine engineers' (N/A, Location N/A). Each card includes a brief description and a 'View Details' button.

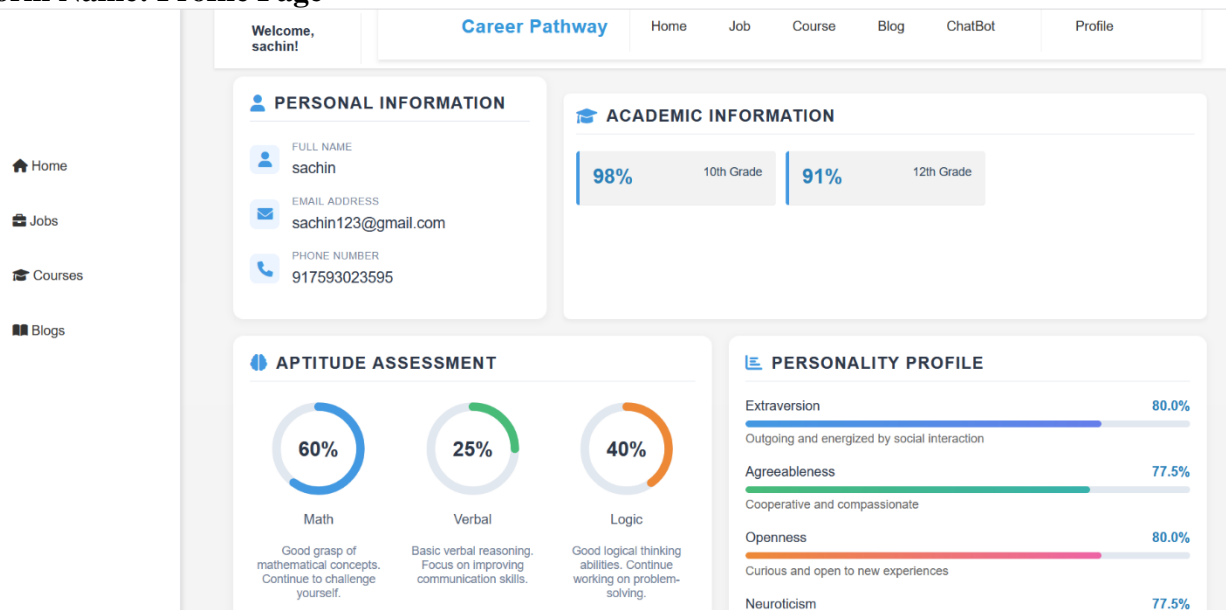
*Fig 4.2.5***Form Name: Course List**

The screenshot displays the 'Course List' page of the Career Pathway application. On the left, a sidebar contains the 'Career Pathway Logo' and a 'Filter by Category' section with checkboxes for Undergraduate (BA, BSc), Post Graduation (M.Sc, M A, MD, Professional), and a 'View Details' button. The main content area features a navigation bar with 'Career Pathway' (active), Home, Job, Course, Blog, ChatBot, and Profile. Below the navigation bar is a section titled 'COURSE' with a search bar and a 'Search' button. The course list is organized into two rows of three cards each. The first row includes 'MCA' (Post Graduation, 25 credits), 'MA English' (Post Graduation, 24 credits), and 'Bsc Maths' (Undergraduate, 36 credits). The second row includes 'Bsc Physics' (Undergraduate, 24 credits), 'Bsc Chemistry' (Undergraduate, 36 credits), and 'MBA' (Post Graduation, 24 credits). Each card provides a description of the course and a 'View Details' button.

Fig 4.2.6

Form Name: Blogs*Fig 4.2.7***Form Name: Psychometric test***Fig 4.2.8*

Form Name: Profile Page



The Profile Page form displays user information and assessment results. It includes a sidebar with navigation links: Home, Jobs, Courses, and Blogs. The main content area is divided into four sections: Personal Information, Academic Information, Aptitude Assessment, and Personality Profile.

Personal Information:

- Full Name: sachin
- Email Address: sachin123@gmail.com
- Phone Number: 917593023595

Academic Information:

- 10th Grade: 98%
- 12th Grade: 91%

Aptitude Assessment:

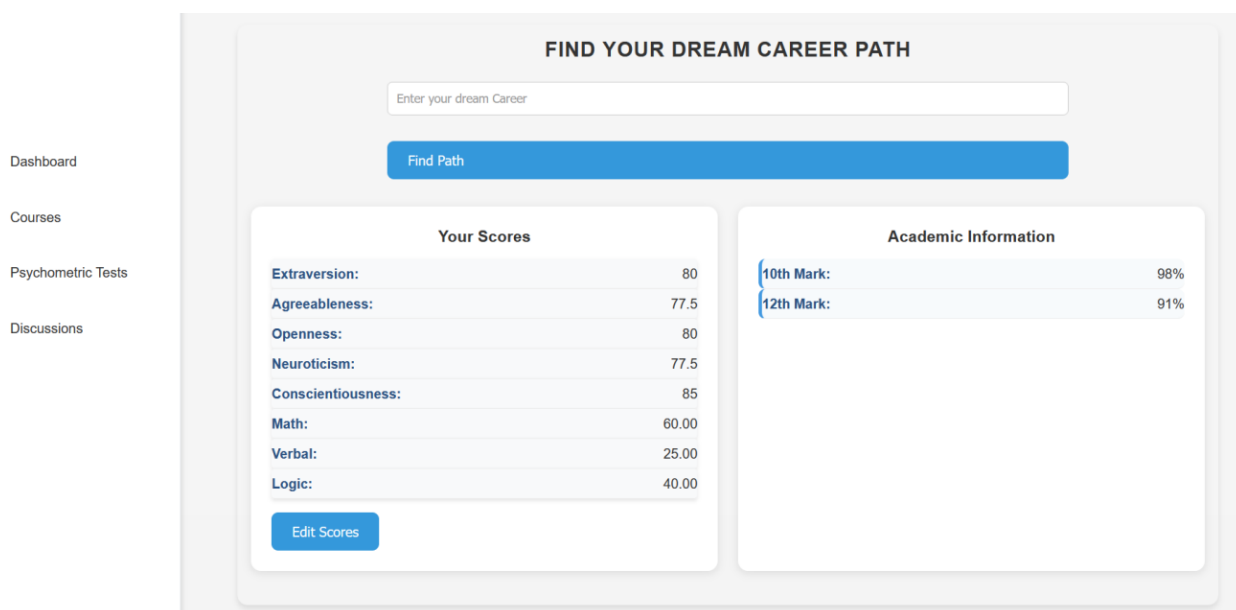
- Math: 60% (Good grasp of mathematical concepts. Continue to challenge yourself.)
- Verbal: 25% (Basic verbal reasoning. Focus on improving communication skills.)
- Logic: 40% (Good logical thinking abilities. Continue working on problem-solving.)

Personality Profile:

- Extraversion: 80.0% (Outgoing and energized by social interaction)
- Agreeableness: 77.5% (Cooperative and compassionate)
- Openness: 80.0% (Curious and open to new experiences)
- Neuroticism: 77.5%

Fig 4.2.9

Form Name: Career Route map



The Career Route map form is titled "FIND YOUR DREAM CAREER PATH". It includes a search bar for "Enter your dream Career" and a "Find Path" button. Below the search bar, there are two main sections: "Your Scores" and "Academic Information".

Your Scores:

Extraversion:	80
Agreeableness:	77.5
Openness:	80
Neuroticism:	77.5
Conscientiousness:	85
Math:	60.00
Verbal:	25.00
Logic:	40.00

Academic Information:

10th Mark:	98%
12th Mark:	91%

There is an "Edit Scores" button at the bottom of the "Your Scores" section.

Fig 4.2.10

Form Name: Career Recommendation

Get Career Predictions

Your Career Recommendations

1. Lawyer
2. Event Manager
3. Marketing Manager

Your Assessment Scores:

Extraversion:	80.00	Agreeableness:	77.50
Conscientiousness:	85.00	Neuroticism:	77.50
Openness:	80.00	Math:	60.00
Verbal:	25.00	Logic:	40.00

Fig 4.2.11

4.3 DATABASE DESIGN

A database is a type of structured storage of data intended to support easy access and updates, with first class information security guarantees. Database design is a two-step process, starting with information-level design (obtaining user requirements and designing a database which strikes an impeccable balance between what the users actually need in their system) Autonomous of any Database Management System (DBMS), this step leads to specific DBMS design for system development Next, comes physical-level design looking to the features of the selected DBMS.

4.4.1 MONGODB DATABASE MANAGEMENT SYSTEM (MDBMS)

A key component of structured data organization and management, MongoDB is a NoSQL database management system. MongoDB stores data in the documents, which are JSONlike flexible structures opposed to traditional RDBMS. A record, which documents in number and attributes or fields that makes up that document. Data integrity and dynamic schema support is catered by MongoDB to ensure it scales. MongoDB with its ability to store and represent more complex data structures is a popular choice for modern applications; it scales easily w/ flexibility between diverse data types.

4.4.2 Sanitization

Sanitization is crucial for maintaining the security and integrity of the data exchanged between users and the system. Both the frontend, built with React.js, and the backend, developed using Node.js, implement sanitization strategies to prevent malicious inputs and ensure data consistency. This process is essential for safeguarding the platform from security threats like SQL injection, cross-site scripting (XSS), and other vulnerabilities.

On the **React.js frontend**, sanitization starts with controlled components, where input fields are linked to the component's state. This allows for real-time validation and filtering of user input before it is sent to the server. Controlled components ensure that only valid and clean data is passed to the backend, helping to prevent common errors or attacks like form submission with incorrect data. Additionally, client-side validation plays a key role in sanitization. Through validation libraries like Yup or Formik, the system can enforce specific data formats, such as ensuring valid email addresses, acceptable price ranges for products, or properly formatted delivery addresses. This front-end validation not only enhances user experience but also reduces the load on the server by catching and filtering invalid data early in the process.

4.4.4 Indexing

The index tracks one piece of data or the set of such data, in order of its corresponding value. Organising the index entries allows fast and manual finding of exact or within range matching things. The indexes make it possible to quickly locate information in a database, instead of scanning the database every time the database is used. An index is the destination roadmap to the information in a database. It allows rapid querying and record ordering as well. More than one column in the table can be used for it.

4.5 Database Collections

4.5.1. Manager

Column	Data Type	Description
name	String	The manager's name.
email	String	The manager's email, must be unique.
password	String	The manager's password, stored as a hash.
confirmPass	String	Field to confirm the password (should match password).
role	Number	Role identifier for the manager (e.g., 1 for manager).

4.5.2. User

Column	Data Type	Description
name	String	The user's name.
email	String	The user's email, must be unique.
password	String	The user's password, stored as a hash.
phone	Number	The user's phone number (optional).
role	Number	The user's role (e.g., 0 for general user, 1 for admin).
resetCode	String	Code used for resetting the user's password, default is an empty string.
resetCodeExpiration	String	Expiration time for the reset code, stored as a date string, default is current time.
education	String	The highest level of education completed (e.g., '10', '+2', 'Under Graduate').
courses	String	List of course names associated with the user, stored as an array of strings.
marks_tenthMark	Number	User's score in 10th grade (default is 0).
marks_twelfthMark	Number	User's score in 12th grade (default is 0).
marks_degreeMark	Number	User's score in undergraduate degree (default is 0).
marks_pgMark	Number	User's score in postgraduate degree (default is 0).
createdAt	String	Timestamp of when the record was created, stored as a date string.
updatedAt	String	Timestamp of the most recent update, stored as a date string.

4.5.3. Blog

Column	Data Type	Description
title	String	The title of the blog post, required field.
content	String	The main content of the blog post, required field.
image	String	URL of the blog post image (optional).
author	String	ID of the user who authored the blog post, required field (stored as a reference).
createdAt	String	Timestamp of when the blog post was created, stored as a date string (default is current time).
status	String	Status of the blog post, default is "active" (can be 'active' or 'blocked').

4.5.4. Category

Column	Data Type	Description
name	String	The name of the category, required field, must be unique.
subcategories	String	List of subcategory names associated with the category, stored as a comma-separated string (optional).

4.5.5. Report

Column	Data Type	Description
blogId	String	ID of the blog post being reported, required field (stored as a reference).
reason	String	Reason for reporting the blog, required field (options include 'Violence Content', 'False Information', 'Nudity or Sexual Content', 'Promoting Unwanted Content', 'I Just Don't Like the Post').
reportedBy	String	ID of the user who reported the blog, required field (stored as a reference).
createdAt	String	Timestamp of when the report was created, stored as a date string

		(default is current time).
isBlocked	Boolean	Status indicating whether the reported blog is blocked, default is false.

4.5.6. Course

Column	Data Type	Description
name	String	The short name of the course, required field.
fullName	String	The full name of the course, required field.
description	String	Detailed description of the course, required field, with a minimum length of 10 characters.
eligibility	String	Eligibility criteria for enrolling in the course, required field.
category	String	ID of the category to which the course belongs, required field (stored as a reference).
subcategory	String	Subcategory associated with the course, required field.
job	String	List of job titles related to the course, stored as a comma-separated string (optional).
entrance	String	List of entrance exams related to the course, stored as a comma-separated string (optional).
duration	Number	Duration of the course in months, required field.

4.5.7. Job

Column	Data Type	Description
name	String	The name of the job, required field, must be unique.
description	String	Detailed description of the job, required field, with a minimum length of 10 characters.
eligibility	String	List of eligibility criteria for the job, stored as a comma-separated string (optional).
industry	String	List of industries related to the job, stored as a comma-separated string (optional).
category	String	ID of the category to which the job belongs, stored as a reference (optional).

4.5.8. Premium User

Column Name	Data Type	Constraints & Description
_id	ObjectId	Primary Key (Unique Identifier)
email	String	Required, Unique (User email)
name	String	Required (User's full name)
isPremium	Boolean	Default: false (Indicates if user is a premium member)
paymentStatus	String	Default: 'pending' (Payment status for premium subscription)

4.5.9. Aptitude Question

Column Name	Data Type	Constraints & Description
_id	ObjectId	Primary Key (Unique Identifier)
question	String	Required
trait	String	Required, Must be one of (Math, Verbal, Logic)
options	Array	Embedded array of option objects
options.text	String	Required (Option text)
options.score	Number	Required, Range: 0-100
createdAt	Date	Default: Date.now

4.5.10. PersonalityResult

Column Name	Data Type	Constraints & Description
_id	ObjectId	Primary Key (Unique Identifier)
email	String	Required, Trimmed
scores.extraversion	Number	Required, Range: 0-100
scores.agreeableness	Number	Required, Range: 0-100
scores.conscientiousness	Number	Required, Range: 0-100
scores.neuroticism	Number	Required, Range: 0-100
scores.openness	Number	Required, Range: 0-100
timestamp	Date	Default: Date.now (Submission time)

4.5.11. MockTest

Column Name	Data Type	Constraints & Description
_id	ObjectId	Primary Key (Unique Identifier)
title	String	Required (Mock test title)
description	String	Required (Brief about the test)
duration	Number	Required (Duration in minutes)
totalMarks	Number	Required (Total marks for the test)
numberOfQuestions	Number	Required (Total number of questions)
questions	Array	Embedded array of question objects
questions.questionText	String	Required (Question text)
questions.options	Array	Embedded array of option objects
questions.options.optionText	String	Required (Option text)
questions.options.isCorrect	Boolean	Required (Indicates if the option is correct)
questions.marks	Number	Required (Marks for the question)
questions.steps	Array of Strings	Required (Step-wise explanation for the question)
passingMarks	Number	Required (Minimum marks required to pass)
status	Boolean	Default: true (Indicates if the test is active)
createdAt	Date	Auto-generated timestamp
updatedAt	Date	Auto-updated timestamp

4.5.12. Question

Column Name	Data Type	Constraints & Description
_id	ObjectId	Primary Key (Unique Identifier)
question	String	Required
options	Array	Embedded array of option objects
options.text	String	Required (Option text)

options.score	Number	Required (Score for the option)
trait	String	Required, Must be one of (Neuroticism, Agreeableness, Conscientiousness, Openness, Extraversion)
createdAt	Date	Default: Date.now (Creation timestamp)

4.5.13. Aptitude Result

Column Name	Data Type	Constraints & Description
_id	ObjectId	Primary Key (Unique Identifier)
email	String	Required
scores.math	Number	Required (Math Score)
scores.verbal	Number	Required (Verbal Score)
scores.logic	Number	Required (Logic Score)
timeRemaining	Number	Default: 0 (Tracks remaining time when submitted)
timestamp	Date	Default: Date.now (Submission time)

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

One of the most important procedure used to validate whether computer program is doing what it is supposed to do is known as software testing. Carried out to ensure the software does what it is supposed to do right; works as per specification and complies with the governed requirement. The purpose of validation is to examine and evaluate a software under test against stated criterion. A method (often integrated with techniques such as code inspection and program walkthrough) to test behavior of software (compared with the object in reality) Software validation assures that the software meets user expectations and needs.

Software testing is undertaken based on several principles and objectives like:

1. Testing: The execution of a program to evaluate its compliance with requirements.
2. Requirements Test Case: An excellent test case is one with probability to find out errors that are not uncovered.
3. A good test is one which reveals previously undetected defects.

If a test case is working fine and fulfill its goal, it will be able to find the defect in software. This indicates that the computer program is working correctly and is doing well. Evaluating Computer Program that has three main facets and includes:

1. Correctness assessment
2. Evaluation of implementation efficiency
3. Examination of computational complexity

5.2 TEST PLAN

A test plan is the overall instruction set for performing different types of tests. Think of it as legal tender when testing a computer program, the plan to follow for its evaluation. Software developer write instructions on how to do something and path for the information needed for a program to be interpret by computer machine. They guarantee that all parts of the program work as it intended to. The ITG (Information Technology Group) actually tests and verify that the software has been exercised properly, rather than depending on software developers for that protection.

The testing objectives must be defined and measurable. A good test plan itself must contain quantifiable information about how often failures are coming, what the cost of repair was and the default rate of failures and time demands for full testing. The different levels of testing are generally broadly:

Unit testing

Integration testing

Data validation testing

Output testing

5.2.1 Unit Testing

Unit testing, checks the tiniest component of software design i.e., The Software component or module. The design guide for testing important control paths within the module. For example: This is what configuring and the extent of testing each little sub program. Unit testing that takes a look at how the code works on multiple parts of a program and can be done for free on different portions in the program.

We have to ensure that data is flowing right between various segments within the computer program even before we start any other test. No point in doing all the other checks if the data doesn't get in and out properly. Planning for contingencies when you design something is very important i.e. thinking of what can go wrong and preparing a plan for those failure scenarios. This can imply redirecting or braking from the process.

In order to test the CareerPathway System, it was taking each part individually and dropping in various tests. Design flaws of the modules were found and corrected in some others. Once the instructions for every parts are written, it gets checked and executed as a whole separately. Eliminated the cruft, and made sure no bits are misbehaving.

5.2.2 Integration Testing

Integration Testing (a very important process in software development is making programs build together and checking interaction of multiple program components at the same time to find errors). The goal is to then use these tested pieces individually, and piece them together again to form the program as planned. This is full test procedure which deals the program and applies to the program for right functioning and functionality.

Integration testing resolves issues identified and addressed but sometimes bring to light other issues which leads to testing further refinement hanging indefinitely. Each of these components are then merged after being comprehensively reviewed by an independent entity to work together. Also, standardization should be attempted on all programs such to not maintain separate versions from one another.

5.2.3 Validation Testing or System Testing

For the last phase of testing, wherein the system tests as a whole to check the cooperation/behavior of other parts (different kind of instructions or building blocks) together happens. Black Box Testing or System testing is the Testing methodology.

Black Box testing is one of the approaches to verify that the application works as intended. It assists software engineers to find every issue in the program by using different input types. Black Box testing introduces error in functions, interfaces, data access, performance, and initialization/exit process etc. It is a cornerstone of ensuring the software properly implements its required functionality and does what it should do.

5.2.4 Output Testing or User Acceptance Testing

System testing is carried out to evaluate user satisfaction and compliance with what the users have requested from the company. To remain on the same level with end-users in development or modification of a computer program, one need to be in context with them. The connection is made by these elements as follows:

- Input Screen Designs.
- Output Screen Designs.

Different kinds of data are utilized for system testing. Such is the role of data preparation in this phase. When data for the test has been taken, then the system under test is measured by using of the test data. This testing should identify issues and those are corrected according to the procedures. These record of such corrections are kept for reference and enhancements in future. The aim of this approach to ensure the system works properly and analyzes in user requirement organization perspective.

5.2.5 Automation Testing

Automated testing is a process and technique used to check that the software is working as per specification and against standards before it gets in production. It is based in the written instructions that are run by testing tools. UI automation testing means a kind of testing where you use tools to automate the testing. Because scripts are made to automatically take care of these tests for multiple situations, instead on the interactions with individuals clicking through the application to ensure functionality. When the testing of a feature needs to be repeated across multiple computers at once, this adds great value for replicating the tests alongside making sure everything is done perfectly.

5.2.6 Selenium Testing

One of the key tools to automate web based testing, selenium In my opinion it is free and very useful. It happens to wet a major role for web developer because it reduces the hassle off testing. Selenium automation : The use of Selenium to carry out automation testing. In fact, Selenium is not an entity tool; it's bunch tools where each tool has its own task in automation testing world. Application development requires manual testing, but it generally becomes tedious and repetitious. Jason Huggins, a ThoughtWorks employee came up with solution for these challenges by creating an automation of testing routines instead manual processes. He started off with a tool called the JavaScriptTestRunner to enable automated testing of websites, and in 2004 they relaunched the product as Selenium.

Example:

Test Case 1

Code :

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class step_definition {
    WebDriver driver=null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette","C:\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver=new FirefoxDriver();
        driver.manage().window().maximize();

    }

    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:5173/login");
        Thread.sleep(2000);

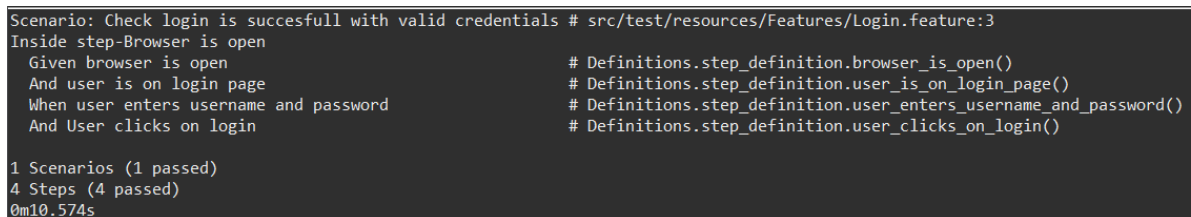
    }
}
```

```
@When("user enters username and password")
public void user_enters_username_and_password() throws Throwable{
    driver.findElement(By.id("emailid")).sendKeys("sachin@gmail.com");
    driver.findElement(By.id("passwords")).sendKeys("Abhi@123");
}

@When("User clicks on login")
public void user_clicks_on_login() {
    driver.findElement(By.id("login")).click(); }

}
```

Screenshot



```
Scenario: Check login is succesfull with valid credentials # src/test/resources/Features/Login.feature:3
Inside step-Browser is open
  Given browser is open # Definitions.step_definition.browser_is_open()
  And user is on login page # Definitions.step_definition.user_is_on_login_page()
  When user enters username and password # Definitions.step_definition.user_enters_username_and_password()
  And User clicks on login # Definitions.step_definition.user_clicks_on_login()

1 Scenarios (1 passed)
4 Steps (4 passed)
0m10.574s
```

Test Report

Test Case 1					
Project Name: Career Pathway					
Login Test Case					
Test Case ID: Test_1			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 15/03/2025		
Module Name: Login Module			Test Executed By : Ms. Merin Chacko		
Test Title : User Login			Test Execution Date: 15/03/2025		
Description: User has a valid email/password					
Pre-Condition :User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : sachin@gmail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
4					
Post-Condition: User is validated with database and successfully logs into Homepage.					

Test Case 2:**Code:**

```
package Definitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class step_definition {
    WebDriver driver = null;

    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette", "C:\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }

    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:5173/login");
        Thread.sleep(2000);
    }

    @When("user click on sign up")
    public void click_on_signup() throws Throwable {
        driver.findElement(By.id("signup")).click();
    }
}
```

```
}

@When("user enters details")
public void user_enters_details() {
    driver.findElement(By.id("name")).sendKeys("Abhi");
    driver.findElement(By.id("email")).sendKeys("abhi@gmail.com");
    driver.findElement(By.id("password")).sendKeys("Abhi@123");
    driver.findElement(By.id("confirmpassword")).sendKeys("Abhi@123");
}

@When("User clicks on sign up")
public void user_clicks_on_signup() {
    driver.findElement(By.id("signup")).click();
}
}
```

Screenshot:

```
Scenario: signup # src/test/resources/Features/career.feature:3
Inside step-Browser is open
  Given browser is open # Definitions.step_definition.browser_is_open()
  When user is on login page # Definitions.step_definition.user_is_on_login_page()
  And user click on sign up # Definitions.step_definition.click_on_signup()
  And user enters details # Definitions.step_definition.user_enters_details()
  And User clicks on sign up # Definitions.step_definition.user_clicks_on_signup()

1 Scenarios (1 passed)
5 Steps (5 passed)
0m11.984s
```

Test report

Test Case 2					
Project Name: Career Pathway					
Registration					
Test Case ID: Test_2			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 14/03/2025		
Module Name: Add blog Module			Test Executed By : Ms. Merin Chacko		
Test Title : Sign Up			Test Execution Date: 14/03/2025		
Description: registration of new user					
Pre-Condition :					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Click on Sign up		User	User successfully sign upped	Pass
3	Provide valid details	Name: Abhi Email:Abhi@gmail.com Password: Abhi@123 Confirmpassword: Abhi@123	Can register as new		
4	Then User clicks on signup button				
Post-Condition: User can now successfully login.					

Test Case 3:**Code:**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class step_definition {

    WebDriver driver=null;

    @Given("browser is open")
    public void browser_is_open() {

        System.out.println("Inside step-Browser is open");

        System.setProperty("webdriver.gecko.marionette","C:\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");

        driver=new FirefoxDriver();

        driver.manage().window().maximize();

    }

    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {

        driver.navigate().to("http://localhost:5173/login");

        Thread.sleep(2000);

    }

    @When("user enters username and password")
    public void user_enters_username_and_password() throws Throwable{

        driver.findElement(By.id("emailid")).sendKeys("sachin@gmail.com");

        driver.findElement(By.id("passwords")).sendKeys("Abhi@123");

    }

}
```

```
}
```

```
@When("User clicks on login")
public void user_clicks_on_login() {
    driver.findElement(By.id("login")).click();
}
```

```
@Then("user is in dashboard")
public void user_is_in_dashboard() throws Exception {
    driver.navigate().to("http://localhost:5173/home");
    Thread.sleep(2000);
}
```

```
@Then("User clicks on profile")
public void user_clicks_on_profile() {
    driver.findElement(By.id("profile")).click();
}
```

```
@Then("user clicks on blog")
public void user_clicks_on_blog() {
    driver.findElement(By.id("addblog")).click();
}
```

```
@Then("user enters the title,content")
public void user_enters_the_title_content() {
    driver.findElement(By.id("text")).sendKeys("hi its me");
    driver.findElement(By.id("content")).sendKeys("writing blog");
}
```

```
@Then("user clicks addblog")
public void user_clicks_addblog() {
```

```
    driver.findElement(By.id("submit")).click();
```



```
}  
  
}
```

Screenshot:

```
Scenario: Check login is succesfull with valid credentials # src/test/resources/Features/Login.feature:3  
  Inside step-Browser is open  
    Given browser is open # Definitions.step_definition.browser_is_open()  
    And user is on login page # Definitions.step_definition.user_is_on_login_page()  
    When user enters username and password # Definitions.step_definition.user_enters_username_and_password()  
    And User clicks on login # Definitions.step_definition.user_clicks_on_login()  
    Then user is in dashboard # Definitions.step_definition.user_is_in_dashboard()  
    Then User clicks on profile # Definitions.step_definition.user_clicks_on_profile()  
    Then user clicks on blog # Definitions.step_definition.user_clicks_on_blog()  
    And user enters the title,content # Definitions.step_definition.user_enters_the_title_content()  
    Then user clicks addblog # Definitions.step_definition.user_clicks_addblog()  
  
1 Scenarios (1 passed)  
9 Steps (9 passed)  
0m14.818s
```

Test report

Test Case 3					
Project Name: Career Pathway					
Add Blog Test Case					
Test Case ID: Test_3			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 15/03/2025		
Module Name: Add blog Module			Test Executed By : Ms. Merin Chacko		
Test Title : Add Blog			Test Execution Date: 15/03/2025		
Description: User can add blogs					
Pre-Condition : User has to be logged in					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : sachin@gmail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
4	Then User clicks on profile				
5	Then user clicks on add-blog And user enters the title,content	Text : HI Content:my first blog	User Should be able to add Blog with text and content	Blog Added	pass
7	Then user clicks addblog				
Post-Condition: User success fully added blog to the database with username .					

Test Case 4:**Code**

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class step_definition {

    WebDriver driver=null;

    @Given("browser is open")

    public void browser_is_open() {

        System.out.println("Inside step-Browser is open");

        System.setProperty("webdriver.gecko.marionette","C:\\\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");

        driver=new FirefoxDriver();

        driver.manage().window().maximize();

    }

    @And("user is on login page")

    public void user_is_on_login_page() throws Exception {

        driver.navigate().to("http://localhost:5173/login");

        Thread.sleep(2000);

    }

    @When("user enters username and password")

    public void user_enters_username_and_password() throws Throwable{
```

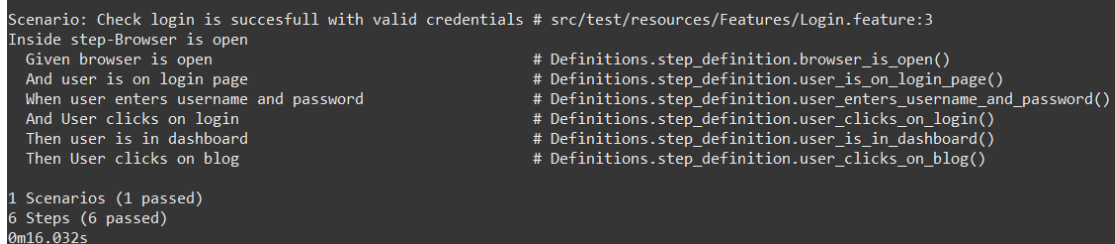
```
        driver.findElement(By.id("emailid")).sendKeys("sachin@gmail.com");
        driver.findElement(By.id("passwords")).sendKeys("Abhi@123");
    }

    @When("User clicks on login")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }

    @Then("user is in dashboard")
    public void user_is_in_dashboard() throws Exception {
        driver.navigate().to("http://localhost:5173/home");
        Thread.sleep(2000);
    }

    @Then("User clicks on blog")
    public void user_clicks_on_blog() {
        driver.findElement(By.id("viewblog")).click();
    }
}
```

Screenshot:



```
Scenario: Check login is succesfull with valid credentials # src/test/resources/Features/Login.feature:3
Inside step-Browser is open
  Given browser is open # Definitions.step_definition.browser_is_open()
  And user is on login page # Definitions.step_definition.user_is_on_login_page()
  When user enters username and password # Definitions.step_definition.user_enters_username_and_password()
  And User clicks on login # Definitions.step_definition.user_clicks_on_login()
  Then user is in dashboard # Definitions.step_definition.user_is_in_dashboard()
  Then User clicks on blog # Definitions.step_definition.user_clicks_on_blog()

1 Scenarios (1 passed)
6 Steps (6 passed)
0m16.032s
```

Test Report

Test Case 4					
Project Name: Career Pathway					
View Blog Test Case					
Test Case ID: Test_4			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 15/03/2025		
Module Name: View Blog Module			Test Executed By : Ms. Merin Chacko		
Test Title : View Blog			Test Execution Date: 15/03/2025		
Description: All the blogs should be visible					
Pre-Condition :User to be logged in					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : sachin@gmail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
4	User clicks on blog	Text : HI Content:my first blog	Blogs should be visible to the user	Blogs are visible	pass
Post-Condition: User can report the blog if needed					

Test Case 5

Code

```
import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import io.cucumber.java.en.And;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;


public class step_definition {

    WebDriver driver = null;


    @Given("browser is open")

    public void browser_is_open() {

        System.out.println("Inside step-Browser is open");

        System.setProperty("webdriver.gecko.marionette", "C:\\\\Users\\\\abhid\\\\eclipse-
workspace\\\\aaa1234\\\\src\\\\test\\\\resources\\\\drivers\\\\geckodriver.exe");

        driver = new FirefoxDriver();

        driver.manage().window().maximize();

    }


    @And("user is on login page")

    public void user_is_on_login_page() throws Exception {

        driver.navigate().to("http://localhost:5173/login");
```

```
        Thread.sleep(2000);

    }

    @When("user enters username and password")

    public void user_enters_username_and_password() throws Throwable {

        driver.findElement(By.id("emailid")).sendKeys("sachin@gmail.com");

        driver.findElement(By.id("passwords")).sendKeys("Abhi@123");

    }

    @When("User clicks on login")

    public void user_clicks_on_login() {

        driver.findElement(By.id("login")).click();

    }

    @Then("user is in dashboard")

    public void user_is_in_dashboard() throws Exception {

        driver.navigate().to("http://localhost:5173/home");

        Thread.sleep(2000);

    }

    @Then("User clicks on courses")

    public void user_clicks_on_courses() {

        driver.findElement(By.id("viewcourse")).click();

    }
```

```
@Then("user is in coursepage")

public void user_is_in_coursepage() throws Exception {

    driver.navigate().to("http://localhost:5173/Ucourselist");

    Thread.sleep(2000);

}

@When("user enters coursename")

public void user_enters_coursename() throws Throwable {

    driver.findElement(By.id("searchin")).sendKeys("MCA");

}

@Then("user click on search")

public void user_click_on_search() {

    driver.findElement(By.id("search")).click();

}

}
```

Screenshot:

```
Scenario: Check view course is succesfull with valid credentials # src/test/resources/Features/Login.feature:3
Inside step-Browser is open
  Given browser is open # Definitions.step_definition.browser_is_open()
  And user is on login page # Definitions.step_definition.user_is_on_login_page()
  When user enters username and password # Definitions.step_definition.user_enters_username_and_password()
  And User clicks on login # Definitions.step_definition.user_clicks_on_login()
  Then user is in dashboard # Definitions.step_definition.user_is_in_dashboard()
  Then User clicks on courses # Definitions.step_definition.user_clicks_on_courses()
  Then user is in coursepage # Definitions.step_definition.user_is_in_coursepage()
  When user enters coursename # Definitions.step_definition.user_enters_coursename()
  Then user click on search # Definitions.step_definition.user_click_on_search()

1 Scenarios (1 passed)
9 Steps (9 passed)
0m15.679s
```


Test report

Test Case 5					
Project Name: Career Pathway					
Search Course					
Test Case ID: Test_5			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 15/03/2025		
Module Name: Add blog Module			Test Executed By : Ms. Merin Chacko		
Test Title : Search Course			Test Execution Date: 15/03/2025		
Description: User can view and search course					
Pre-Condition :User has to be logged in					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : sachin@gmail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
5	user clicks on course		User Should be able to view and search all the course	Course discovered successfully	pass
6	User clicks on searchbox, Then user clicks on search	Text : MCA			
Post-Condition: User can go through all the details of the course					

Test Case 6:**Code**

```
package Definitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class step_definition {
    WebDriver driver = null;

    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette", "C:\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }

    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:5173/login");
        Thread.sleep(2000);
    }

    @When("user enters username and password")
    public void user_enters_username_and_password() throws Throwable {
        driver.findElement(By.id("emailid")).sendKeys("sachin123@gmail.com");
    }
}
```

```
        driver.findElement(By.id("passwords")).sendKeys("Abhi@123");  
    }  
  
    @When("User clicks on login")  
    public void user_clicks_on_login() {  
        driver.findElement(By.id("login")).click();  
    }  
  
    @And("clicks on Tests")  
    public void clicks_on_psychometric_tests() throws Exception{  
        Thread.sleep(2000);  
        driver.findElement(By.id("tests")).click(); // Replace with actual locator  
    }  
  
    @Then("clicks on manual career assignment")  
    public void clicks_on_manual_career_assignment() {  
        driver.findElement(By.id("Manual Career Assessment")).click();  
    }  
  
    @Then("enter scores")  
    public void enter_scores() {  
        // Locate the score input fields and enter scores  
        WebElement extraversionInput = driver.findElement(By.id("extraversion"));  
        WebElement agreeablenessInput = driver.findElement(By.id("agreeableness"));  
        WebElement opennessInput = driver.findElement(By.id("openness"));  
        WebElement neuroticismInput = driver.findElement(By.id("neuroticism"));  
        WebElement conscientiousnessInput =  
driver.findElement(By.id("conscientiousness"));  
        WebElement mathInput = driver.findElement(By.id("math"));  
        WebElement verbalInput = driver.findElement(By.id("verbal"));  
        WebElement logicInput = driver.findElement(By.id("logic"));
```

```

// Enter scores
extraversionInput.sendKeys("75");
agreeablenessInput.sendKeys("80");
opennessInput.sendKeys("70");
neuroticismInput.sendKeys("60");
conscientiousnessInput.sendKeys("85");
mathInput.sendKeys("90");
verbalInput.sendKeys("78");
logicInput.sendKeys("88");
}

@Then("click on Get career prediction")
public void click_on_get_career_prediction() {
    // Locate the Get Career Predictions button and click it
    WebElement getCareerPredictionButton =
driver.findElement(By.xpath("//button[text()='Get Career Predictions']"));
    getCareerPredictionButton.click();
}
}

```

Screenshot

```

Scenario: Check view Manual career recommendation # src/test/resources/Features/career.feature:3
Inside step-Browser is open
  Given browser is open # Definitions.step_definition.browser_is_open()
  And user is on login page # Definitions.step_definition.user_is_on_login_page()
  When user enters username and password # Definitions.step_definition.user_enters_username_and_password()
  And User clicks on login # Definitions.step_definition.user_clicks_on_login()
  And clicks on Tests # Definitions.step_definition.clicks_on_psychometric_tests()
  And clicks on manual career assignment # Definitions.step_definition.clicks_on_manual_career_assignment()
  Then enter scores # Definitions.step_definition.enter_scores()
  And click on Get career prediction # Definitions.step_definition.click_on_get_career_prediction()

1 Scenarios (1 passed)
8 Steps (8 passed)
0m16.345s

```

Test report

Test Case 6					
Project Name: Career Pathway					
Career Recommendation through Manual Input					
Test Case ID: Test_6			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 16/03/2025		
Module Name: Career Recommend			Test Executed By : Ms. Merin Chacko		
Test Title : Manual career recommendation			Test Execution Date: 16/03/2025		
Description: User can view and search course					
Pre-Condition :User has to be logged in					
St ep	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fai l)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : Sachin123@gmail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
5	user clicks on Tests		User Should be able to view the suggest career	Career suggestion successfully	pass
6	User clicks on Manual career assignment And enter the values	Extraversion = 75, agreeablenes s = 80, neuroticism =60, openness =70,			
7	user clicks on career prediction				
Post-Condition: User see the suggested career based on their aptitude and personality score					

Test Case : 7**Code**

```
package Definitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
public class step_definition {
    WebDriver driver = null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette", "C:\\\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }
    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:5173/login");
        Thread.sleep(2000);
    }
    @When("user enters username and password")
    public void user_enters_username_and_password() throws Throwable {
        driver.findElement(By.id("emailid")).sendKeys("sachin123@gmail.com");
        driver.findElement(By.id("passwords")).sendKeys("Abhi@123");
    }
    @When("User clicks on login")
```

```

public void user_clicks_on_login() {
    driver.findElement(By.id("login")).click();
}

@And("clicks on Tests")
public void clicks_on_psychometric_tests() throws Exception{
    Thread.sleep(2000);
    driver.findElement(By.id("tests")).click(); // Replace with actual locator
}

@Then("clicks on manual career assignment")
public void clicks_on_manual_career_assignment() {
    driver.findElement(By.id("Manual Career Assessment")).click();
}

@And("click on fetch scores")
public void clicks_on_fetch_scores() {
    driver.findElement(By.id("fetch score")).click();
}

@Then("click on Get career prediction")
public void click_on_get_career_prediction() {
    // Locate the Get Career Predictions button and click it

    WebElement                getCareerPredictionButton                =
driver.findElement(By.xpath("//button[text()='Get Career Predictions']"));
    getCareerPredictionButton.click();
} }

```

Screenshot:

```

Scenario: Check view career recommendation # src/test/resources/Features/career.feature:3
Inside step-Browser is open
  Given browser is open                # Definitions.step_definition.browser_is_open()
  And user is on login page            # Definitions.step_definition.user_is_on_login_page()
  When user enters username and password # Definitions.step_definition.user_enters_username_and_password()
  And User clicks on login             # Definitions.step_definition.user_clicks_on_login()
  And clicks on Tests                  # Definitions.step_definition.clicks_on_psychometric_tests()
  And clicks on manual career assignment # Definitions.step_definition.clicks_on_manual_career_assignment()
  And click on fetch scores            # Definitions.step_definition.clicks_on_fetch_scores()
  And click on Get career prediction    # Definitions.step_definition.click_on_get_career_prediction()

1 Scenarios (1 passed)
8 Steps (8 passed)
0m14.518s

```

Test report

Test Case 7					
Project Name: Career Pathway					
Career Recommendation					
Test Case ID: Test_7			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 16/03/2025		
Module Name: Career Recommend			Test Executed By : Ms. Merin Chacko		
Test Title : career recommendation (Auto fetch)			Test Execution Date: 16/03/2025		
Description: User can view and search course					
Pre-Condition :User has to be logged in					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fai l)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : Sachin123@g mail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
5	user clicks on Tests	Extraversion= 80, Openness=75,	User Should be able to view the suggest career	Career suggestion successfully	pass
6	User clicks on Manual career assignment	Click Manual assignment			
7	Click on fetch scores And click career prediction	Click career prediction			
Post-Condition: User see the suggested career based on their aptitude and personality score					

Test case: 8**Code**

```
package Definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class step_definition {

    WebDriver driver = null;

    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette", "C:\\Users\\abhid\\eclipse-
workspace\\aaa1234\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }

    @And("user is on login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://localhost:5173/login");
        Thread.sleep(2000);
    }

    @When("user enters username and password")
    public void user_enters_username_and_password() throws Throwable {
        driver.findElement(By.id("emailid")).sendKeys("sachin123@gmail.com");
        driver.findElement(By.id("passwords")).sendKeys("Abhi@123");
    }

    @When("User clicks on login")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }
}
```

```

    }
    @And("clicks on Tests")
    public void clicks_on_psychometric_tests() throws Exception{
        Thread.sleep(2000);
        driver.findElement(By.id("tests")).click(); // Replace with actual locator
    }
    @When("clicks on Dream career")
    public void clicks_on_dream_career() throws InterruptedException {
        Thread.sleep(2000); // Ensure the element is visible before clicking
        WebElement dreamCareerButton = driver.findElement(By.id("Dream career")); //
Check the actual ID in the HTML
        dreamCareerButton.click();
    }
    @When("user enters career")
    public void user_enters_career() {
        driver.findElement(By.id("PathBox")).sendKeys("Teacher");
    }
    @When("Find path")
    public void find_path() {
        driver.findElement(By.id("FindPath")).click();
    }
}

```

Screenshot:

```

Scenario: Dream Career                                     # src/test/resources/Features/career.feature:3
Inside step-Browser is open
  Given browser is open                                   # Definitions.step_definition.browser_is_open()
  And user is on login page                               # Definitions.step_definition.user_is_on_login_page()
  When user enters username and password                  # Definitions.step_definition.user_enters_username_and_password()
  And User clicks on login                               # Definitions.step_definition.user_clicks_on_login()
  And clicks on Tests                                    # Definitions.step_definition.clicks_on_psychometric_tests()
  And clicks on Dream career                             # Definitions.step_definition.clicks_on_dream_career()
  And user enters career                                 # Definitions.step_definition.user_enters_career()
  And Find path                                          # Definitions.step_definition.find_path()

1 Scenarios (1 passed)
8 Steps (8 passed)
0m16.381s

```

Test report

Test Case 8					
Project Name: Career Pathway					
Career Recommendation through Manual Input					
Test Case ID: Test_8			Test Designed By: Abhidarsh K S		
Test Priority(Low/Medium/High): High			Test Designed Date: 16/03/2025		
Module Name: Career Recommend			Test Executed By : Ms. Merin Chacko		
Test Title : career recommendation (Auto fetch)			Test Execution Date: 16/03/2025		
Description: Dream Career Route map					
Pre-Condition :User has to be logged in					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fai l)
1	Navigate to login page		Login form should be displayed	Login form is displayed	Pass
2	Provide valid Email	Email : Sachin123@gmail.com	User should be able to login	User logs in	Pass
3	Provide valid password	Password: Abhi@123			
5	user clicks on Tests		User Should be able to view the suggest career	Career suggestion successfully	pass
6	User clicks on Dream Career enters the career	Text : Teacher			
7	user clicks on Find path	Click Find Path			
Post-Condition: User can see the path towards their dream career					

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation this is when the planned system goes from being a paper system into an actual live & operational system entity. Trust and confidence in users, is to the success of any thing at all system will of. This is critical stage which gets high on user training and informative stuff. In reality, the actual change usually happens while doing user training. it means to put a new system in place after its constructing point of view not only, but also likely as completion of implementation a blueprint to a usable form.

Implementation is the act converting from old way of doing things to new deployment, whether it just replaces it or changes incrementally. To ensure that the process is done right is very critical to build a efficient system that fits in with the organization. System implementation includes:

- Meticulous planning.
- Assessment of the existing system and its constraints.
- Designing methods to facilitate the transition.

6.2 IMPLEMENTATION PROCEDURES

Software implementation means install where software should be and check that it is hammer steel as expected. In some firms, the individual unconnected from employing the software may govern and approve the project development of the software. At least to start, the software will be met with some skepticism and needs to be addressed to prevent a huge resistance.

Key Steps To Make Transition Successful:

- Benefits Realization: Users have to get over that the new software is better than the present one, so they can trust it.
- User Training: This lets the users feel comfortable & confident enough to use the application by training them.

In order to check its output, you need to ensure that the server program is up and running on the server. If server is down, there is not expected result to get.

6.2.1 User Training

User training is one of the most important task to make people aware that, how they will kickstart and adapt to new system. Important for user comfort and confidence with the system. More complex the system gets, less to do with training as such. Issues in user training are diverse — we can learn how to enter data, interact with queries against databases, using tools that help you generate reports or do essential things. Focusing on its goal is to be able to do

this on their own by teaching individuals the knowledge and skills they would need best in using the system.

6.2.2 Training on the Application Software

After explaining the basics of computer skills, we take individuals through a new software to use. This training will educate users on whole system from, screen navigation, to resources on how to find help in case they come across errors with the solution or procedures. The training is designed to provide users or defined user groups with the understanding and ability to leverage the system or subsystem appropriately. Keep in mind that the training could be delivered to groups of users and persons in various roles throughout the organization depending on what is essential to make real sense of all this.

6.2.3 System Maintenance

Keep a software running is a difficult problem in the field of software development. Phase of software life cycle in which the software is in return for its life-critical uses and it just works as it should do in maintenance phase. A system that functions properly, a living system that requires constant tending in order to function at peak efficiency after deployment. It is a vital part of the dev process as this takes care of the system's capability to evolve according to its changing environments. Maintenance is more than just finding and fixing code errors. You fix the stability and effectiveness of your system in many ways from maintenance.

6.2.4 Hosting

The CareerPathway (hosted on Render and Netlify due to their reliability, performance for smooth development stack integration). Render for backend server deploys, Netlify deploys front end interfaces for customers, admin and district operation managers. Together these platforms deliver a frictionless, scalable user experience for the website visitors.

Render

Render is the cloud-based hosting platform that makes it easier to deploy end-to-end applications (websites or native App) best for triggers server side application like my backend API. For Node backends that Render powers, a trusted place to deploy the high-performance API calls into our frontends on Netlify. Here are the key reasons to choose Render:

- **Automatic Deployments:** Render automatically builds and deploys from GitHub repos whenever there are new commits preventing that our backend server doesn't go down.
- **SSL Support:** Render provides the free SSL certificate for the secure data transfer of transactions that happen between our server and client application ramps.
- **Scalability:** Render comes in handy for expanding deployments so that the backend scales

and can process high traffic, as well as our ever growing API requests.

Hosting Process on Render (Backend)

Step 1- Create a Render Account

Start an Account and Sign In to Render (render.com)

2: GitHub Repository Connection

Add the GitHub repo with backend code to be linked.

Step 3 — Create a New Web Service

- Select “New web service” → linked repository
- Configure the environment
- Build Command: `npm install`
Start Command: `node index.js`
- Add the environment variables required
- Enable auto-deploy to your CI for every new commit

Step 4: Set Environment Variables:

Define environment variables for database connection , API Keys and so on configurations.

Step 6: Deploy the Service

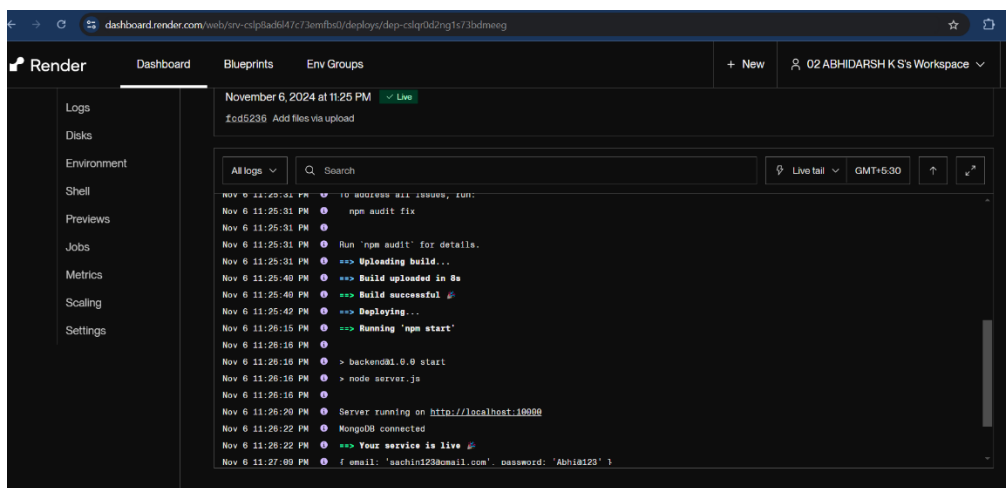
Build the build and deploy the backend server live on an URL from Render.

Step 6: Test & Monitor

Ensure the backends performance and error logs through logged on Render’s built-in analytics.

Hosted Link: <https://pathway-1-frontend.onrender.com>

Screenshot



*Fig 6.2.4.1***Hosting Process on Render (Frontend)****Procedure 1: Create Render Account**

Sign Up and Log in to Render (render.com)

Procedure 2: Link your GitHub Repository

Connect the repository with the backend code

Procedure 3 — Adding a new web service:

- Select New Web Service and the connected repo
- Configure the environment
- Build Command: npm install

Start Command node index.js

- Insert required Environment Variables
- Activate CI auto-deploy on every new commit

Procedure 4 to configure Environment Variables :

Deploy the API with connection environment variables. API KEYS configurations

Procedure 5. Deploy Your API

This means that Render takes care of the build and deployment, so it's actually the backend web server is now running on a secure URL.

Procedure 6: Test And Monitor

Monitor the performance, logs and events for backends behaviour with built-in logs and analytics by Render.

Hosted Website: Render

Hosted Link: <https://pathway-1-frontend.onrender.com>

Hosted Link QR Code

Screenshot

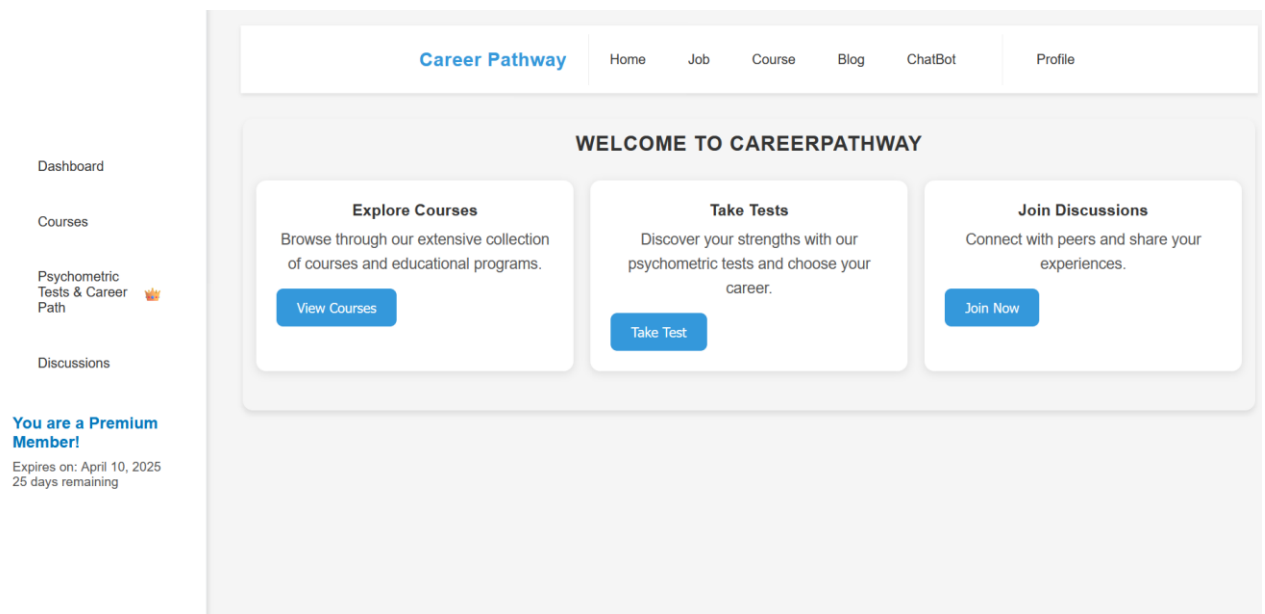


Fig 6.2.4.2

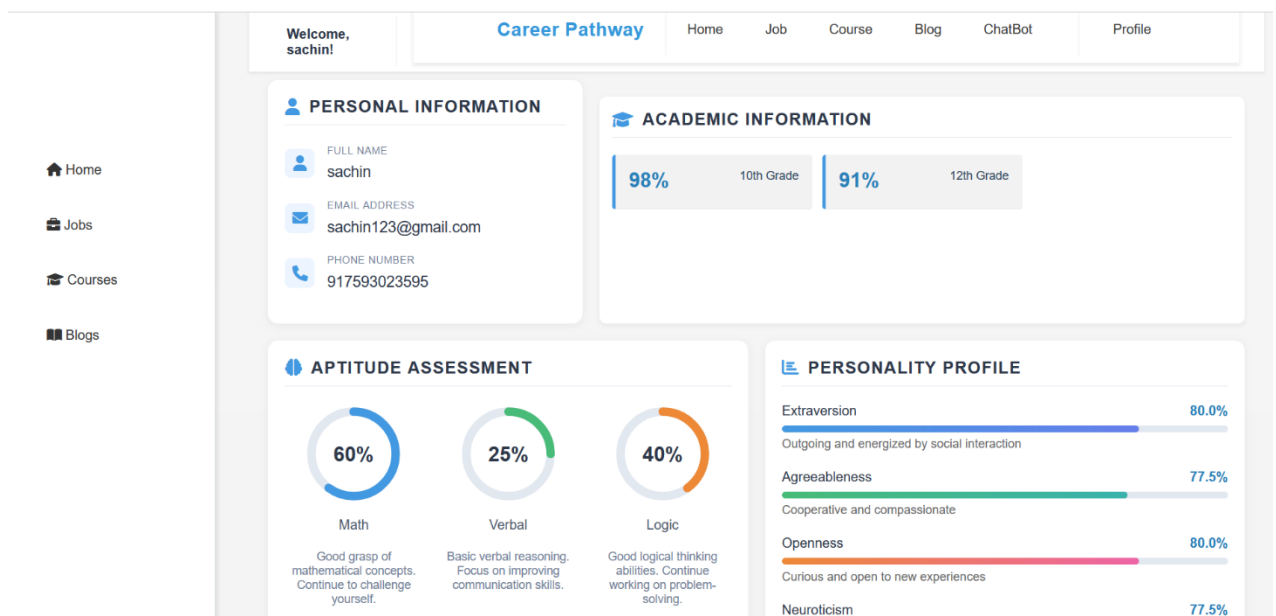


Fig 6.2.4.3

The screenshot displays the 'Career Pathway' website interface. On the left, a sidebar contains the 'Career Pathway Logo' and a 'Filter by Category' section. This section has two main categories: 'Undergraduate' (with sub-options BA and BSc) and 'Post Graduation' (with sub-options M.Sc, M A, MD, and Professional). The main content area is titled 'COURSE' and features a search bar with the placeholder 'Search Course...' and a 'Search' button. Below the search bar, there are six course cards arranged in a 2x3 grid. Each card includes the course name, a brief description, a category tag (e.g., 'Post Graduation' or 'Undergraduate'), a duration icon (clock), and a 'View Details' button.

Course Name	Description	Category	Duration	Action
MCA	The Master of Computer Applications (MCA) is a two-year postgraduate degree program that teaches students about computer applications and softwar...	Post Graduation	25	View Details
MA English	A Master of Arts (MA) in English is a postgraduate degree that involves the study of English literature, linguistics, and language. It is a popular language...	Post Graduation	24	View Details
Bsc Maths	BSc Mathematics, or BSc Maths, is an undergraduate degree program that focuses on the study of mathematical principles, theories, and...	Undergraduate	36	View Details
Bsc Physics	A Bachelor of Science (BSc) in Physics is a three-year undergraduate degree program that studies the fundamental principles of the natural world. The...	Undergraduate	24	View Details
Bsc Chemistry	A Bachelor of Science (BSc) in Chemistry is an undergraduate degree program that studies the properties, structure, composition, reactions, and...	Undergraduate	36	View Details
MBA	MBA stands for Master of Business Administration, a professional postgraduate degree that focuses on business administration. It's a two-year program th...	Post Graduation	24	View Details

Fig 6.2.4.3

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The Career Pathway project is an inclusive platform that caters the requirement of students in their transition phase of career and academic life. And with equally new features like ML recommendation and AI route map to dream careers on this Students are more effectively enable to achieve careers. The system inform students en masse with information on courses, career pathways and learning opportunities to make more informed decisions based on real data. This project aims to provide an easy access platform for students also in terms of supporting, blogging and interacting career-related information.

Career Pathway, major features like its course management, job listing (using ML), AI route map recommendations and feedback loop/service, platform for blogs all provide value, easy and seamless consumer experience. The platform provides strong suite of tools on the admin & manager side to manage content, reports, mock-test/psychometric tests moderation and user interactions. This provides a smooth experience for all users and ensures the quality and relevance of the things the platform is providing.

7.2 FUTURE SCOPE

For future development and development, The Career Pathway project has major room to expand. A key area for future development, is the use of advanced data analytics to give students more granular career view and education outcome information. Finally, the more information about the student profile such as his hobbies and other extra-curricular activities etc. gives in addition is more valuable in knowing about and recommending careers. Career Pathway can make its resources more student centric & effective by examining the student engagement data and providing personalized course/career recommendations in sync with emerging industry trends based on student's interests.

Further, the addition of interactive features such as virtual mentorship sessions and career-oriented webinars would serve users with relevant information and networking possibilities straight on the platform. Additional content depth might be added through partnerships with educational establishments and industry influencers alike, allowing students to learn from the experts in particular areas.

Real-time notifications, bookmarking and content curation tools are other features we recommend to the platform as it grows and makes the actual experience better (usability, engagement). As these advancements occur, Career Pathway can update its model to fit the changing educational and career landscape to continue serving as a relevant resource to students who are figuring out their futures.

CHAPTER 8

BIBLIOGRAPHY

BOOKS

- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley Professional.
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
- Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill.
- MongoDB Documentation (2024). Used for data storage and management in the CareerPathway project.
- ReactJS Documentation (2024). Reference for front-end development of CareerPathway using ReactJS.
- Node.js Documentation (2024). Guide for server-side development using Node.js.

WEBSITES

- MERN Stack Guide - MongoDB, Express.js, React.js, Node.js. Retrieved from:
- NPM (Node Package Manager).
- Career Guide
- Career360
- Shiksha

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.jsx

```
import { useState } from "react";
import axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import styles from "../styles.module.css";
import { getAuth, signInWithPopup, GoogleAuthProvider } from "firebase/auth";
import { firebaseApp } from "../../firebase"; // Ensure firebaseApp is initialized

const auth = getAuth(firebaseApp);
const googleProvider = new GoogleAuthProvider();

const Login = () => {
  const [data, setData] = useState({ email: "", password: "" });
  const [error, setError] = useState("");
  const navigate = useNavigate();

  const handleChange = ({ currentTarget: input }) => {
    setData({ ...data, [input.name]: input.value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      const url = "http://localhost:8080/login";
      const { data: res } = await axios.post(url, data); // Destructure response

      if (res.token) {
        const { role } = res.data;

        localStorage.setItem("token", res.token); // Store the JWT token
        localStorage.setItem("userRole", role); // Store the user's role
      }
    } catch (error) {
      setError(error.message);
    }
  };

  return (
    <div>
      <div>
        <input type="text" value={data.email} onChange={handleChange} />
        <input type="password" value={data.password} onChange={handleChange} />
      </div>
      <button type="button" onClick={handleSubmit}>Login</button>
      <div>{error}</div>
    </div>
  );
};
```



```
switch (role) {
  case 1:
    navigate("/home");
    break;
  case 2:
    navigate("/manager");
    break;
  default:
    navigate("/home"); // Default route
}
}
} catch (error) {
  if (
    error.response &&
    error.response.status >= 400 &&
    error.response.status <= 500
  ) {
    setError(error.response.data.message);
  }
};

// Function to handle Google login
const signInWithGoogle = () => {
  signInWithPopup(auth, googleProvider)
    .then((result) => {
      const user = result.user;
      const fields = {
        name: user.providerData[0].displayName,
        email: user.providerData[0].email,
      };

      axios
        .post("http://localhost:8080/authWithGoogle", fields)
        .then((res) => {
```

```
    if (!res.data.error) {
      localStorage.setItem("token", res.data.token);
      navigate("/home");
    } else {
      setError(res.data.msg);
    }
  });
});
.catch((error) => {
  setError(error.message);
});
};

return (
  <div className={styles.login_container}>
    <div className={styles.login_form_container}>
      <div className={styles.left}>
        <form className={styles.form_container} onSubmit={handleSubmit}>
          <h1>Login to Your Account</h1>
          <input
            type="email"
            placeholder="Email"
            name="email"
            id="emailid"
            onChange={handleChange}
            value={data.email}
            required
            className={styles.input}
          />
          <input
            type="password"
            placeholder="Password"
            name="password"
            id="passwords"
```

```
        onChange={handleChange}
        value={data.password}
        required
        className={styles.input}
      />
      { /* Forgot Password Button */}
      <div style={{ textAlign: "right", marginTop: "5px" }}>
        <Link
          to="/forgotpassword"
          style={{ textDecoration: "none", color: "#007bff" }}
        >
          Forgot Password?
        </Link>
      </div>
      {error && <div className={styles.error_msg}>{error}</div>}
      <button id="login" type="submit" className={styles.green_btn}>
        Sign In
      </button>
      <button
        type="button"
        className={styles.google_login_button}
        onClick={signInWithGoogle}
      >
        Sign In with Google
      </button>
    </form>
  </div>
  <div className={styles.right}>
    <h1>New Here?</h1>
    <Link to="/signup">
      <button type="button" className={styles.white_btn}>
        Sign Up
      </button>
    </Link>
```

```
    </div>
  </div>
</div>
);
};

export default Login;
```

Server.js

```
const express = require("express");
const mongoose = require("mongoose");
const dotenv = require("dotenv");
const jwt = require("jsonwebtoken");
const bodyParser = require("body-parser");
const bcrypt = require("bcrypt");
const cors = require("cors");
const nodemailer = require("nodemailer");
const User = require("../models/RegisterModel.js");
const app = express();

// Import Routes
const CourseRoute = require("../routes/course.js");
const JobRoute = require("../routes/job.js");
const Viewjob = require("../routes/viewjobs.js");
const addmanager = require("../routes/addmanager.js");
const Viewcourse = require("../routes/viewcourses.js");
const DelCourseRoute = require("../routes/delcourse.js");
const DelJobRoute = require("../routes/deljob.js");
const ViewManager = require("../routes/viewmanager.js");
const ViewUserPro = require("../routes/uprofile.js");
const Ueditpro = require("../routes/uprofileupdate.js");
const categoryRoutes = require("../routes/categoryRoutes.js");
const SearchCourse = require("../routes/searchCourse.js");
const blogRoutes = require("../routes/blogRoutes.js");
```

```
const managerModel = require("./models/AddManager.js");
const reportRoutes = require("./routes/reportRoutes.js");
const ViewReportRoutes = require("./routes/Reportroute.js");
// Middleware for CORS
app.use(
  cors({
    origin: ["http://localhost:5173"],
    methods: ["GET", "POST", "PUT", "DELETE"],
    credentials: true,
  })
);

// Load environment variables
dotenv.config();

// MongoDB Connection
mongoose
  .connect(process.env.MONGODB_URI || "mongodb://localhost:27017/pathway")
  .then(() => console.log("MongoDB connected"))
  .catch((err) => console.error("MongoDB connection error:", err));

// Middleware to parse JSON
app.use(bodyParser.json());
app.use(express.json());
app.use("/uploads", express.static("uploads")); // Serve uploaded images

// Blog routes
app.use("/api/blogs", blogRoutes);

// Routes for user authentication and password reset
app.post("/authWithGoogle", async (req, res) => {
  const { name, email } = req.body;
  try {
    const existingUser = await User.findOne({ email });
```

```
if (!existingUser) {
  const result = await User.create({
    name,
    email,
    password: " ",
    isAdmin: false,
  });
  const token = jwt.sign(
    { email: result.email, id: result._id },
    process.env.JSON_WEB_TOKEN_SECRET_KEY,
    { expiresIn: "1h" }
  );
  return res.status(200).send({
    user: result,
    token,
    msg: "User Registered and Logged in Successfully!",
  });
} else {
  const token = jwt.sign(
    { email: existingUser.email, id: existingUser._id },
    process.env.JSON_WEB_TOKEN_SECRET_KEY,
    { expiresIn: "1h" }
  );
  return res.status(200).send({
    user: existingUser,
    token,
    msg: "User Logged in Successfully!",
  });
}
} catch (error) {
  console.log(error);
  return res
    .status(500)
    .send({ msg: "An error occurred", error: error.message });
}
```

```
    }  
  });  
  
  // Signup and login routes  
  app.post("/signup", async (req, res) => {  
    const { name, email, password } = req.body;  
    if (!name || !email || !password) {  
      return res.status(400).json({ msg: "Please fill all the fields." });  
    }  
    if (password.length < 6) {  
      return res  
        .status(400)  
        .json({ msg: "Password must be at least 6 characters long." });  
    }  
    try {  
      const existingUser = await User.findOne({ email });  
      if (existingUser) {  
        return res.status(400).json({ msg: "User already exists" });  
      }  
      const newUser = new User({ name, email, password, role: 1 });  
      await newUser.save();  
      res.status(200).json({ msg: "User registered successfully" });  
    } catch (error) {  
      console.error(error);  
      res.status(500).json({ msg: "Server error", error: error.message });  
    }  
  });  
  
  app.post("/login", async (req, res) => {  
    const { email, password } = req.body;  
    console.log(req.body);  
  
    if (!email || !password) {  
      return res.status(400).json({ message: "All fields are required" });  
    }  
  });
```

```
}

try {
  const user = await User.findOne({ email });
  const manager = await managerModel.findOne({ email });

  // If neither user nor manager is found
  if (!user && !manager) {
    return res.status(400).json({ message: "Invalid email or password." });
  }

  let currentUser, role;

  if (manager) {
    const isMatch = await bcrypt.compare(password, manager.password);
    if (!isMatch) {
      return res.status(401).json({ message: "Invalid email or password" });
    }
    currentUser = manager;
    role = manager.role;
  } else if (user) {
    if (user.password !== user.password) {
      return res.status(401).json({ message: "Invalid email or password" });
    }
    currentUser = user;
    role = user.role;
  }

  // Generate JWT token
  const token = jwt.sign(
    { _id: currentUser._id, email: currentUser.email },
    process.env.JSON_WEB_TOKEN_SECRET_KEY,
    { expiresIn: "1h" }
  );
}
```



```
);

return res.status(200).json({
  message: "Login Successful",
  data: {
    email: currentUser.email,
    role: role,
    name: currentUser.name,
    userId: currentUser._id,
  },
  token: token,
});
} catch (error) {
  console.error("Login error:", error);
  res
    .status(500)
    .json({ message: "Something went wrong.", error: error.message });
}
});

// Password reset routes
app.post("/forgotpassword", async (req, res) => {
  const { email } = req.body;
  const user = await User.findOne({ email });
  if (!user) {
    return res
      .status(400)
      .send({ error: true, msg: "Email id not registered" });
  }
  const resetCode = Math.floor(1000 + Math.random() * 9000).toString();
  user.resetCode = resetCode;
  user.resetCodeExpiration = Date.now() + 3600000;
  await user.save();
  const transporter = nodemailer.createTransport({
```

```
    service: "gmail",
    auth: { user: process.env.EMAIL, pass: process.env.PASSWORD },
  });
const mailOptions = {
  from: process.env.EMAIL,
  to: user.email,
  subject: "Password Reset Code",
  text: `Your password reset code is ${resetCode}`,
};
transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    return res.status(500).send({ error: true, msg: "Error sending email" });
  } else {
    return res
      .status(200)
      .send({ error: false, msg: "Verification code sent to your email." });
  }
});
});

app.post("/verifycode", async (req, res) => {
  const { email, code } = req.body;
  const user = await User.findOne({
    email,
    resetCode: code,
    resetCodeExpiration: { $gt: Date.now() },
  });
  if (!user) {
    return res
      .status(400)
      .send({ error: true, msg: "Invalid or expired code" });
  }
  res.status(200).send({ error: false, msg: "Code verified" });
});
```

```
app.post("/resetpassword", async (req, res) => {  
  const { email, password } = req.body;  
  const user = await User.findOne({ email });  
  if (!user) {  
    return res.status(400).send({ error: true, msg: "Email not found" });  
  }  
  user.password = password;  
  user.resetCode = undefined;  
  user.resetCodeExpiration = undefined;  
  await user.save();  
  res.status(200).send({ error: false, msg: "Password has been reset" });  
});
```

```
// Use routes
```

```
app.use("/course", CourseRoute);  
app.use("/job", JobRoute);  
app.use("/viewjob", Viewjob);  
app.use("/addmanager", addmanager);  
app.use("/viewcourse", Viewcourse);  
app.use("/delcourse", DelCourseRoute);  
app.use("/deljob", DelJobRoute);  
app.use("/viewmanager", ViewManager);  
app.use("/vuprofile", ViewUserPro);  
app.use("/updateProfile", Ueditpro);  
app.use("/category", categoryRoutes);  
app.use("/", SearchCourse);  
app.use("/blog", blogRoutes);  
app.use("/report", reportRoutes);  
app.use('/viewreport',ViewReportRoutes);
```

```
// Example route for categories and subcategories
```

```
app.get("/categories/:categoryId/subcategories", async (req, res) => {  
  const { categoryId } = req.params;
```

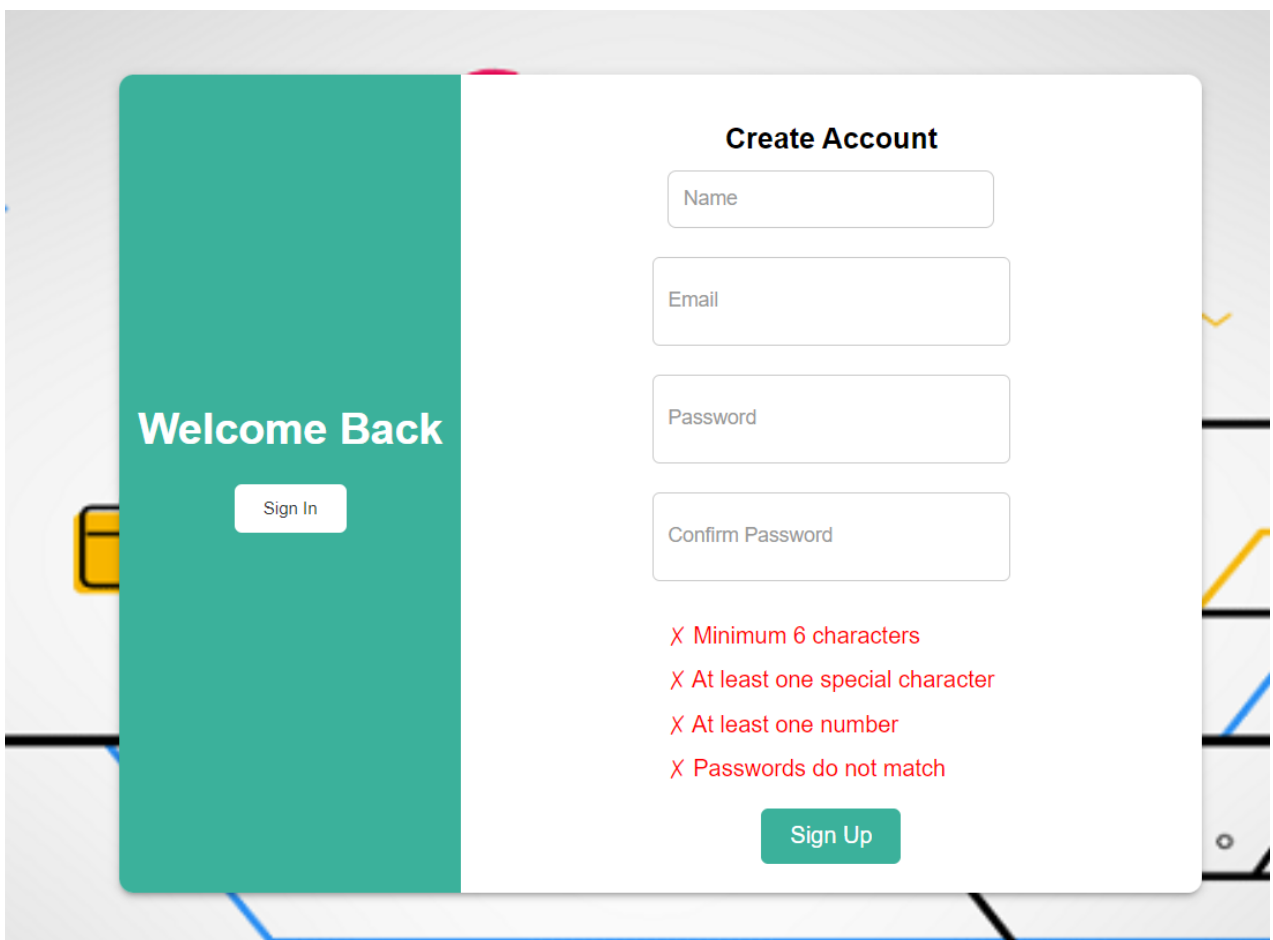
```
// Fetch subcategories based on categoryId from your database
});

// Start the server
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});

app.get('/api/test', (req, res) => res.json({ message: 'API is working' }));
```

9.2 Screen Shots

9.2.1 Sign up page:



Welcome Back

[Sign In](#)

Create Account

X Minimum 6 characters

X At least one special character

X At least one number

X Passwords do not match

[Sign Up](#)

9.2.2 Login page

The login page features a central form with the title "Login to Your Account". It includes input fields for "Email" and "Password", a link for "Forgot Password?", a "Sign In" button, and a "Sign In with Google" option. To the right, a teal sidebar contains the text "New Here?" and a "Sign Up" button.

9.2.3 Job List Page

The job list page has a sidebar on the left with the "Career Pathway Logo" and a "Filter by Category" section. The filter section includes checkboxes for "Undergraduate" (with sub-options BA and BSc) and "Post Graduation" (with sub-options M.Sc, M A, MD, and Professional). The main content area has a navigation bar with links: "Career Pathway", "Home", "Job", "Course", "Blog", "ChatBot", and "Profile". Below the navigation bar is a section titled "AVAILABLE JOB OPPORTUNITIES" containing three job cards: "Data Scientist", "Software development", and "Marine engineers". Each card includes a brief description, a category tag (Undergraduate, Post Graduation, or N/A), a location indicator (Location N/A), and a "View Details" button.

9.2.4 Course list page

Career Pathway

Home Job Course Blog ChatBot Profile

COURSE

Search Course... Search

Filter by Category

☐ Undergraduate

☐ BA

☐ BSc

☐ Post Graduation

☐ M.Sc

☐ M.A

☐ M.D

☐ Professional

MCA

The Master of Computer Applications (MCA) is a two-year postgraduate degree program that teaches students about computer applications and software development.

Post Graduation 25

[View Details](#)

MA English

A Master of Arts (MA) in English is a postgraduate degree that involves the study of English literature, linguistics, and language. It is a popular language...

Post Graduation 24

[View Details](#)

Bsc Maths

BSc Mathematics, or BSc Maths, is an undergraduate degree program that focuses on the study of mathematical principles, theories, and...

Undergraduate 36

[View Details](#)

Bsc Physics

A Bachelor of Science (BSc) in Physics is a three-year undergraduate degree program that studies the fundamental principles of the natural world. The...

Undergraduate 24

Bsc Chemistry

A Bachelor of Science (BSc) in Chemistry is an undergraduate degree program that studies the properties, structure, composition, reactions, and...

Undergraduate 36

MBA

MBA stands for Master of Business Administration, a professional postgraduate degree that focuses on business administration. It's a two-year program th...

Post Graduation 24

9.2.5 Blog page

BLOG POSTS

Search blogs...

Balancing Health And Study

Midhun

Balancing health and study is crucial for overall well-being and academic success. Good physical health, achieved through a balanced diet, regular exercise, and adequate sleep, provides energy and improves concentration, allowing for better learning and retention. Mental health is equally important; managing stress, taking breaks, and staying organized can enhance focus and reduce burnout. Creating a structured study routine with periodic breaks supports effective learning while avoiding exhaustion. Additionally, practicing mindfulness techniques, like meditation, can help keep the mind calm and focused. Prioritizing both health and study ensures a sustainable and productive approach to academic life.

Violence Content Report

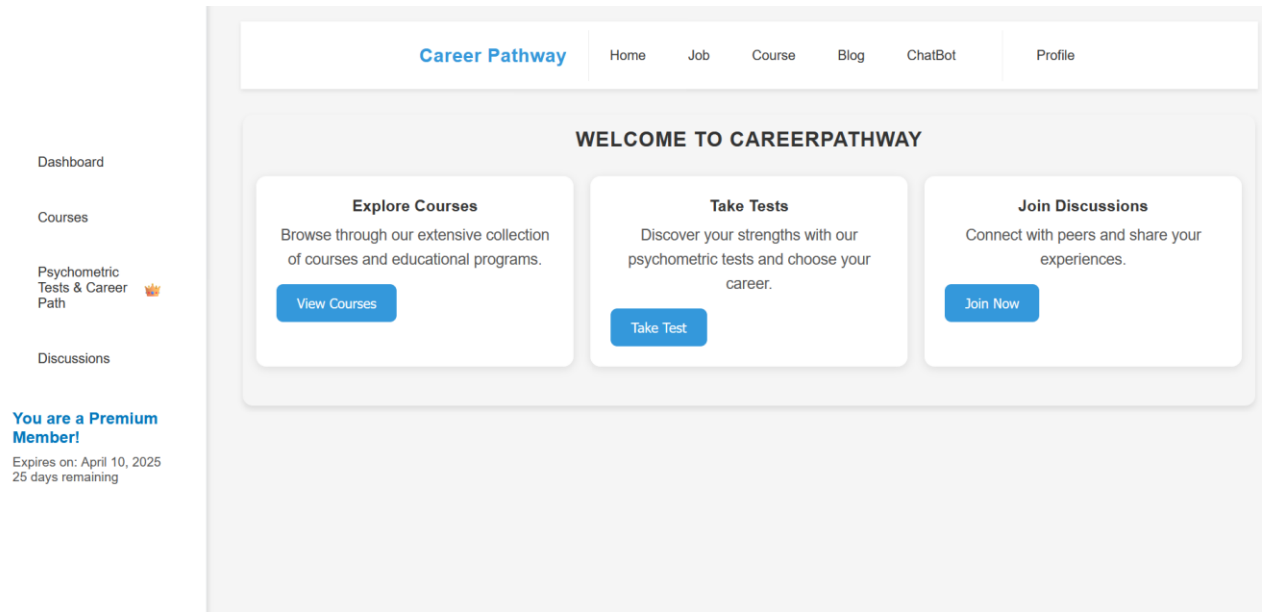
Time Management Tips For Students: Maximize Productivity And Reduce Stress

sachin

Time management is a vital skill for students aiming to balance academics, extracurricular activities, and social life. Good time management not only boosts productivity but also reduces stress, making it easier to meet goals without feeling overwhelmed. Here are some practical strategies for effective time management:

1. Set Clear Goals Define specific, achievable goals for each week or month. Whether it's completing assignments, studying for exams, or engaging in hobbies, clear goals give direction and motivation.
2. Prioritize Tasks with the "Eisenhower Matrix" Use the Eisenhower Matrix to categorize tasks

9.2.6 Home Page



Career Pathway Home Job Course Blog ChatBot Profile

WELCOME TO CAREERPATHWAY

Explore Courses
Browse through our extensive collection of courses and educational programs.
[View Courses](#)

Take Tests
Discover your strengths with our psychometric tests and choose your career.
[Take Test](#)

Join Discussions
Connect with peers and share your experiences.
[Join Now](#)

Dashboard

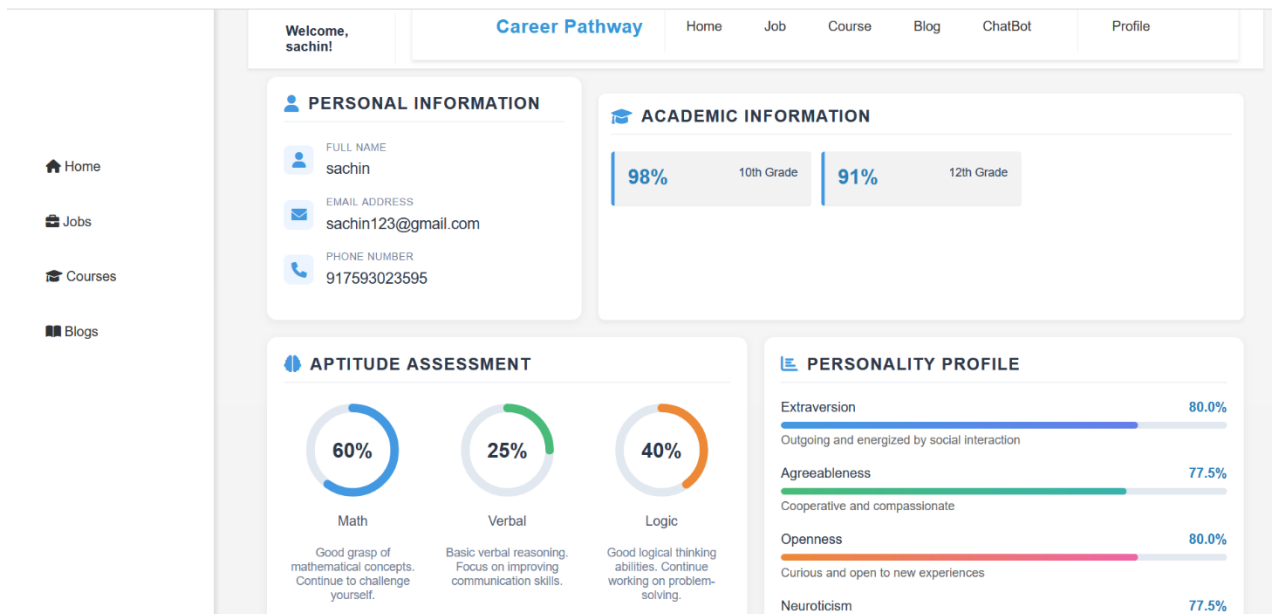
Courses

Psychometric Tests & Career Path 👑

Discussions

You are a Premium Member!
Expires on: April 10, 2025
25 days remaining

9.2.7 Profile Page



Welcome, sachin! **Career Pathway** Home Job Course Blog ChatBot Profile

PERSONAL INFORMATION

FULL NAME
sachin

EMAIL ADDRESS
sachin123@gmail.com

PHONE NUMBER
917593023595

ACADEMIC INFORMATION

98% 10th Grade **91%** 12th Grade

APTITUDE ASSESSMENT

60% Math
Good grasp of mathematical concepts. Continue to challenge yourself.

25% Verbal
Basic verbal reasoning. Focus on improving communication skills.

40% Logic
Good logical thinking abilities. Continue working on problem-solving.

PERSONALITY PROFILE

Extraversion 80.0%
Outgoing and energized by social interaction

Agreeableness 77.5%
Cooperative and compassionate

Openness 80.0%
Curious and open to new experiences

Neuroticism 77.5%

Home

Jobs

Courses

Blogs

9.2.8 Career Path page

Dashboard
Courses
Psychometric Tests
Discussions

FIND YOUR DREAM CAREER PATH

Find Path

Your Scores

Extraversion:	80
Agreeableness:	77.5
Openness:	80
Neuroticism:	77.5
Conscientiousness:	85
Math:	60.00
Verbal:	25.00
Logic:	40.00

Edit Scores

Academic Information

10th Mark:	98%
12th Mark:	91%

9.2.9 Career Recommendation page

Home
Jobs
Courses
Blogs
Tests

Get Career Predictions

Your Career Recommendations

- Lawyer
- Event Manager
- Marketing Manager

Your Assessment Scores:		Extraversion:	80.00	Agreeableness:	77.50
Conscientiousness:	85.00	Neuroticism:	77.50	Openness:	80.00
Math:	60.00	Verbal:	25.00	Logic:	40.00

9.3 Git Log

```
commit 88c579ba3870c36ed7e189c53a3a110b53598bfe
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Mar 16 21:58:13 2025 +0530
```

Add axios dependency to package.json and package-lock.json

```
commit 7e6c8021d48711b5f0c6f2a5909a2e6679ef2796
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Mar 16 21:49:42 2025 +0530
```

ML started

```
commit 3e8fee62270b0db4a0461476c7a3e0abce07400f
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Mar 16 11:48:46 2025 +0530
```

ML stopped 2

```
commit 02a98c3c71b91dd8455e8f50370feac40f0a89fb
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Mar 16 11:45:50 2025 +0530
```

ml stoped temp

```
commit 5f86dbcc46cb7bdb9cc06f1dfa3b3fcb780cc90d
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Mar 16 11:08:21 2025 +0530
```

cors

```
commit 72e4c8a5741f05290c045459d04ebfc5686fb733
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sat Mar 15 23:00:15 2025 +0530
```

redesigned header

commit d012bdc9c6337d1d553eeab0a832065fe826df7b

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sat Mar 15 22:43:40 2025 +0530

Final report generated

commit 87715f5a5a238fa902510f538c3d6d2e2f45d4fa

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sat Mar 15 22:34:37 2025 +0530

Report generated in dream career

commit e281fc3b3384603cbe8467aaab418edccec077e5

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Fri Mar 14 16:04:57 2025 +0530

Timer added to aptitude test

commit d4bc1a2d78d3c78f23b3742c5929581b2de7bb35

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Fri Mar 14 10:23:54 2025 +0530

Dream Career Fetching details also correctly fetched

commit 3207368c7e2442e6b89a8d51216d5961280b0760

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Fri Mar 14 10:19:28 2025 +0530

Manual Career Fetched aptitude correctly

commit 271142fee613436a197dfbc1fbc979c19d35007

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Mar 13 16:36:32 2025 +0530

profile design and fetching scores

commit 91a86a8a555329de1661df4e686d4145a46ab453

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Mar 13 16:15:09 2025 +0530

28days Timelimit for Premium

commit 471bb5238b3ee5bbdb82453d2f5a0c10286a880a

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Mar 13 16:06:30 2025 +0530

Premium memberships is added

commit 389a37f0fd3ec5eb76c637677442a329633cb00d

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Mar 13 10:21:19 2025 +0530

Aptitude test added without time with traits

commit e59ccaf670f68c156347e12ed8ee7ed779d8bd5f

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Mar 11 12:18:53 2025 +0530

re-updated "\${import.meta.env.VITE_URL}"

commit e78893dcc62435bbc975576c324dff01716a1aa5

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Mar 11 09:47:02 2025 +0530

Design DreamCareer and data for ml added

commit 02646341282b2f1f36a1103ce366c2495b5b94f9

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Mar 10 14:31:15 2025 +0530

Career pathway to dreamcareer

commit 1d84685297b77309c680980084c5101519bbb29c

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Mar 10 09:28:21 2025 +0530

dreamcareer, fetching and manually edited

commit 14d9eae511bb6abd6d765b6e76fe8cbfb903786f
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Mar 4 09:53:29 2025 +0530

Changes for Manual testing success

commit cca180b1eab7b13f603f8c19a411fd874fac2edf
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Mar 3 21:27:34 2025 +0530

Manual career Updated, 3 careers are listed

commit 107ad72b58548b8f8dbb0107c7cc2f5569933515
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Mar 3 11:22:31 2025 +0530

Dream job fetching stage completed

commit 17433f92a66194a97a71852272f2bd2e5f81be1b
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Wed Feb 19 10:03:59 2025 +0530

Score set between 10 and 100

commit 8fefe90b78394f6ee07c867e17acfee54d24dc98
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Wed Feb 19 09:52:30 2025 +0530

aptitude score fetched and displayed in manual career

commit 1c9b418de5910a8c3d15884f4bac459ddcfef72a
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Wed Feb 19 09:37:53 2025 +0530

Fetch personality score into manual career prediction boxes (Multiplied by 2.5)

commit aab8200110faabf35e237560cf4ff761eef1f559
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Wed Feb 19 09:20:02 2025 +0530

Change personality score out of 100

commit 25f0f198dc3caec5427322e37f49f41c96120f08
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Feb 18 15:15:02 2025 +0530

result calculated and fetched correctly in testresult

commit d0220cbebd3c5ec1cc00b77ff2e36e0fc8a1f49
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Feb 18 11:36:46 2025 +0530

Personatlity questions automatically advance to next question

commit 44bd851526846331421680f4c6b5ef5a4b2fe2cb
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Feb 17 09:27:24 2025 +0530

ML 3rd stage, more data need to be added

commit 77c6b3761431e8cc7880eae616af472978b7760f
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Fri Feb 14 09:23:55 2025 +0530

ML second update: prediction using CSV

commit bf4643c7812be026e08936fd9fe15dea3fd3f88b
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Wed Feb 12 21:23:02 2025 +0530

ML primary implementation (flask, but not fetched user score done differently)

commit 0098a2ec62106e21daea9954d7b89e0de0a2d9be
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Feb 11 14:57:05 2025 +0530

Personality score fetched and viewed in user profile

commit fe64ce4a2649e901da0542f18a3419e9bc299f62

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Feb 11 14:11:56 2025 +0530

aptitude result added to profile page

commit b635a544dff196562868f06dad226b093b808362

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Feb 11 13:35:04 2025 +0530

Personality test in user side

commit 38e082844b441d341a874ca25679e5dfab09bf24

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Feb 10 11:55:48 2025 +0530

design delete button on personality test

commit 17a36073ffc54a2dd2b6c6742f52170f2b790c38

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Feb 10 10:14:23 2025 +0530

Personality test interface added

commit dc48c524576694a6eafa2f76326dccdc4b2ec165

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Feb 6 21:25:16 2025 +0530

Admin delete and update test

commit 6d5a047809e556bdf6a0af6075a4e332b0ba6c03

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Feb 5 15:41:13 2025 +0530

Timer added

commit 3bd17f893891f6cceb32a0998afca4bbe2b1f896
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Feb 4 11:29:09 2025 +0530

design in admin and manager

commit 4b5a7fbccf666bff0d0e9d24ec79b960fc7c42ee
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Feb 4 09:44:23 2025 +0530

profile updated

commit 031dc35960e72a8baba1b67e6f5c97c63fec7bb0
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Feb 3 22:22:09 2025 +0530

geminiapi customized

commit 28651bbfeaa0b3ce1d23bcd956f1279092b5c687
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Feb 3 16:38:42 2025 +0530

small changes on design and components

commit ed8dbcf70a79fa36b950ebc713248ac1d22aed4
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Feb 3 09:13:11 2025 +0530

Quiz result saved in database

commit ae89fc2bd026f3eab493cfa5a0ce745750d3bbe4
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Feb 2 23:23:23 2025 +0530

test aptitude added and viewed

commit 6bf82d9f405a052279cb34792c1db728cd7644bc
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Wed Jan 29 12:24:52 2025 +0530

Pages redesigned

commit d903748760daf4f5e74e0932e0e32b846c0d70f1
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Jan 27 11:59:45 2025 +0530

ChatBot_Gemini primary setup success

commit 0dd62ab272ce17cbbd0033972021f56fe70ff747
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Jan 21 14:53:33 2025 +0530

small error, Institute registration

commit 884b2569b84954a9cb3f3584c2d9dec6634ff59b
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Jan 21 14:52:01 2025 +0530

institute Registration

commit 7d31c39378ea312d17bffd15671b4b5a2aeac6b
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Fri Jan 10 09:16:57 2025 +0530

user filter remove

commit 2fd12507a0197a7c4b56c450db0e6cc4c95a0d21
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Nov 11 10:33:38 2024 +0530

admin functionalities

commit 22eea1b1f119f6e8ffaa60de7621127de361125d
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Nov 11 10:24:12 2024 +0530

error manager on job and course

commit 27014ce7081f6e81561751963c79ff6c92ba699a

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Nov 11 10:14:49 2024 +0530

manager

commit 0c4334ef72ce0ea88cb864c30e96590368d1801f

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Nov 11 10:09:18 2024 +0530

manager add job

commit 9f2c3c3ef8f3413c633d4eb819e86fc453b6f1c0

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Nov 11 09:59:17 2024 +0530

error in blog edit and admin add job

commit 1109d4128adbc11df3d5ab01f50424eb17046d07

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sun Nov 10 23:40:14 2024 +0530

manager list

commit 23ad718ad657d4c121594c9807c3de17ff4e1dca

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sun Nov 10 22:05:21 2024 +0530

icons added

commit 5b3199e792a05465bd05cf2a23fb57b5081b04be

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sun Nov 10 22:03:32 2024 +0530

admin auth design

commit e2c33f580aead8da0359cb287bb03f5ee0ead124
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Nov 10 14:26:56 2024 +0530

deployment-a

commit 0d872c5f13c2e8742f44ed5df71a828c804126da
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Thu Nov 7 23:14:52 2024 +0530

deploy part 2

commit 83f907b6dc1be6f3eb8931ad1af5af85f2e9b2dc
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Thu Nov 7 23:03:40 2024 +0530

deploy part 1

commit 5feb9f903a70bd6aa33523be809a467f4e7f6bfd
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Thu Nov 7 21:24:42 2024 +0530

made changes in API URL

commit 06b24a2c22b6d9a381b7ad22772d91c394ef822f
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Nov 5 14:15:05 2024 +0530

blog adding error fixed (Token)

commit 07075ed6a3042d3657196e1fd7b08047e864bdff
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Nov 4 21:49:02 2024 +0530

Admin block blog success

commit 269e6e74129e4baff344edcafb3c860512f4c07
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Nov 4 19:11:49 2024 +0530

Add page for report Reports.jsx

commit b08997ad4de856170658e5ffd2c067473335c064
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sun Nov 3 19:14:45 2024 +0530

report by user success

commit 061e1e9f4dde815aa11764d11e62bd2a075176f2
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sat Oct 26 10:33:04 2024 +0530

Updated filter: filter course and job also added category for job

commit 077e10602de0a4e39076e716be68d20fd04bd900
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sat Oct 26 09:13:11 2024 +0530

Home page to default and blog like and comment button disabled

commit 3c9ea27f1dbc0464e375a2298c8f3d001dcbda79
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Fri Oct 25 10:07:16 2024 +0530

id added to login page

commit 3f81e0302e09306efa0c1fcbc0f05e77e64a2ee2
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 21 12:27:55 2024 +0530

user can edit and delete blog

commit a91fb6116b44e847b99350eea895ce7529bf3581
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Oct 21 00:37:14 2024 +0530

login for manager success

commit eac4d091bd4319a3f853fa7dc6291ba14c63cd96

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Fri Oct 18 11:29:50 2024 +0530

style: blog, search and filter -Button

commit b618bc9b831aabca6af00ff1a98d11d4f4c3e89b

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Oct 17 21:47:52 2024 +0530

Blog added successfully

commit ea1816226fad3bec414c51b7d206d856b9bdd984

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 16 16:04:27 2024 +0530

Filter, subcategory dropdown and header profile logout button updated

commit c8b31d2d327b81378e34060de7b67276ef965d98

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 16 15:09:17 2024 +0530

subcategoey added to courseforms

commit e50fac49324c501172e7a5b24e1b8d7581c95919

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 16 14:08:58 2024 +0530

subcategoey added implemented only on filter

commit dc8ff4936b6dea56038fde7e3165e1cc457e7ef2

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Oct 15 15:30:14 2024 +0530

User courses as component

commit 5985ace196e063de96933d574eb37c1bbe6d0835
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Oct 15 12:02:38 2024 +0530

Search and moth validation

commit 83f6f3208b87ae3424a9cd061cc3c2303d994470
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 14 23:00:12 2024 +0530

searchCourse

commit bbf3ce06260c704d8938916fb9914147f0fe0dd9
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 14 22:11:08 2024 +0530

Filter course

commit cdfc17cd6959cb72d06d7216826cbc044d06aad1
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 14 20:45:14 2024 +0530

category duplication in small and capital letter resolved

commit 3cf67b1ce323bf068017e1797baf6f5427f92fb7
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sat Oct 12 12:25:00 2024 +0530

career path logio + category form validation

commit 254e9232c7cfd249ab5aff33c8fadde5975e1a93
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Sat Oct 12 12:04:25 2024 +0530

usersidebar and useauth for all user pages

commit 9cd58e859a88680571f282aad0536786c13e83e6

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 23:16:46 2024 +0530

userauth function for token removal is added

commit ea30028164372db59ed0dfa36a55a3ee1a0a4715

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 22:43:30 2024 +0530

Userprofile view through login

commit 3e547c1866d5bce0a7fd6436b7a7a634bffb5179

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 15:58:21 2024 +0530

manager edit errors corrected

commit 24a47c404a122fcf4d501249fa3b8f2ff23ebd17

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 12:15:53 2024 +0530

course fullname added

commit 868c817eb4118db7cc947a1c28f0b130fab4cea3

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 11:29:38 2024 +0530

manager view course and job

commit cb03118808a3827543d6a9cd9d7afbfb22e164ae

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 10:34:35 2024 +0530

manager add job

commit 555cad5c304cfe4511b0330c5bc56412a008e77f

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 10:29:57 2024 +0530

manager added course updated

commit 3df2bc7c91de518d1e85ff5bd9c15608abad20e6

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 10:19:43 2024 +0530

job and course foem validation (name contain only Alphabets and period)

commit ccd28f537d6a8584937a14b279cc91cc17f5f440

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 09:52:37 2024 +0530

Validation: job duplication

commit b522afbbcb22c098babdf59245218b922aa9b910

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Wed Oct 9 09:46:12 2024 +0530

Validation in login and Course duplication (CSS in multiple pages(marginleft:250px))

commit 898642e32f956c44fc7fec7d3afba7b8ef4d6222

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Oct 8 21:04:14 2024 +0530

Category added dynamically CSS alignment gone wrong

commit 516af21fd86f1e1c81a0750065159284cf1ca571

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Oct 8 12:00:06 2024 +0530

mcourseform: manager- add course

commit abb5193a954e16636542dfc6eee436178a3d0bbe

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Oct 8 11:28:42 2024 +0530

User Header

commit 865e901d45fc24e285eee6548c3c18007ecd0ce8
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Oct 8 10:29:35 2024 +0530

Edit job

commit 1451ad02faeb46c7a7c8aad69c0376650d5ab64a
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Oct 8 10:01:58 2024 +0530

User Job View

commit f53554209555710b95941f357d09a081c966a076
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Tue Oct 8 09:51:54 2024 +0530

User Course view

commit a44a4ea195f5920b64b0282a17b62e0985542f05
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 22:01:47 2024 +0530

user profile update

commit 3905a682a4cd6b780bd09344b0d4f1b41d7bf45b
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 18:21:05 2024 +0530

View Manager

commit 4c3196e87e43491845f6c58814199ff736fbba40
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 14:29:11 2024 +0530

delect job

commit 62b2d2b58907968f2b6e96b14cfa14dd5cae7f1b
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 10:28:58 2024 +0530

Change Style Job

commit c7ad40ec86fcca9f11f92085a023c1d0253be6c8
Merge: 23d9e31c 5f6c9526
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 10:28:19 2024 +0530

Merge branch 'main' of <https://github.com/AbhiDarsh2001/Pathway>

commit 23d9e31ce0b97fca95807d5f08afde2b496b4874
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 09:54:42 2024 +0530

addmanagerstyle

commit 0b7480cec2cd7ad42433c5ce4af267ef9620d300
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Mon Oct 7 09:47:07 2024 +0530

addmanager

commit 5f6c9526b3d1ae813dea8b6c2e4def22722b4a6d
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Thu Sep 26 22:46:21 2024 +0530

view and delect added for course

commit 5268275b1c3fb810faa4a112559c0cff18e4fdd3
Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>
Date: Thu Sep 26 14:34:25 2024 +0530

registerinbox_mailvalidation

commit 1e143b2788b8edf0fff51e705e0edc301a4bf2bb

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Sep 24 23:09:00 2024 +0530

viewjob

commit 1caf53284019770e6f85ce54d2c63a1cf7972675

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Sep 24 12:18:48 2024 +0530

dashboardstyle

commit aac4d4a1c4f37c5e131275c0d3caf0daf3576fba

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Tue Sep 24 11:15:17 2024 +0530

AddJob

commit 50e582ede69ed341b05f748a990c2bd735ab9cb4

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Sep 23 23:38:02 2024 +0530

addcourse

commit bd4bdcb402e110e5e48971f0a86658a4eeab12da

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Mon Sep 23 11:24:36 2024 +0530

ForgotPassword

commit a4b301e97a19f69e94b11f5f62155bf2020cf682

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sun Sep 22 23:47:00 2024 +0530

signup Validation

commit 8fcb6b1f7a7889d8913c9110ea4d36d855e7e1f4

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Sat Aug 31 23:05:47 2024 +0530

Home_Frontend

commit 25fac3b602f2447bc5e2cf2026b3efcf1ebcc789

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Aug 29 23:10:22 2024 +0530

GoogleLoginFull

commit 83730803d8c0e01792afd02c6803665d487c6aa8

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Aug 29 22:36:41 2024 +0530

googleButton to LoginIn

commit 461b6730f56b0aa0b69a6481bf48a0aa6230517f

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Aug 29 14:39:03 2024 +0530

Project Commit

commit b6f5961f527cc5f59b940c36e578424527b5e7bf

Author: AbhiDarsh2001 <abhidarsh2001@gmail.com>

Date: Thu Aug 29 14:32:33 2024 +0530

Documents Added

commit 885b594c3b688f28b23d97e42da7a200bc817513

Author: AbhiDarsh2001 <149134704+AbhiDarsh2001@users.noreply.github.com>

Date: Mon Aug 12 21:59:36 2024 +0530

Abstract

9.4 Certificates

Fig 9.3.1



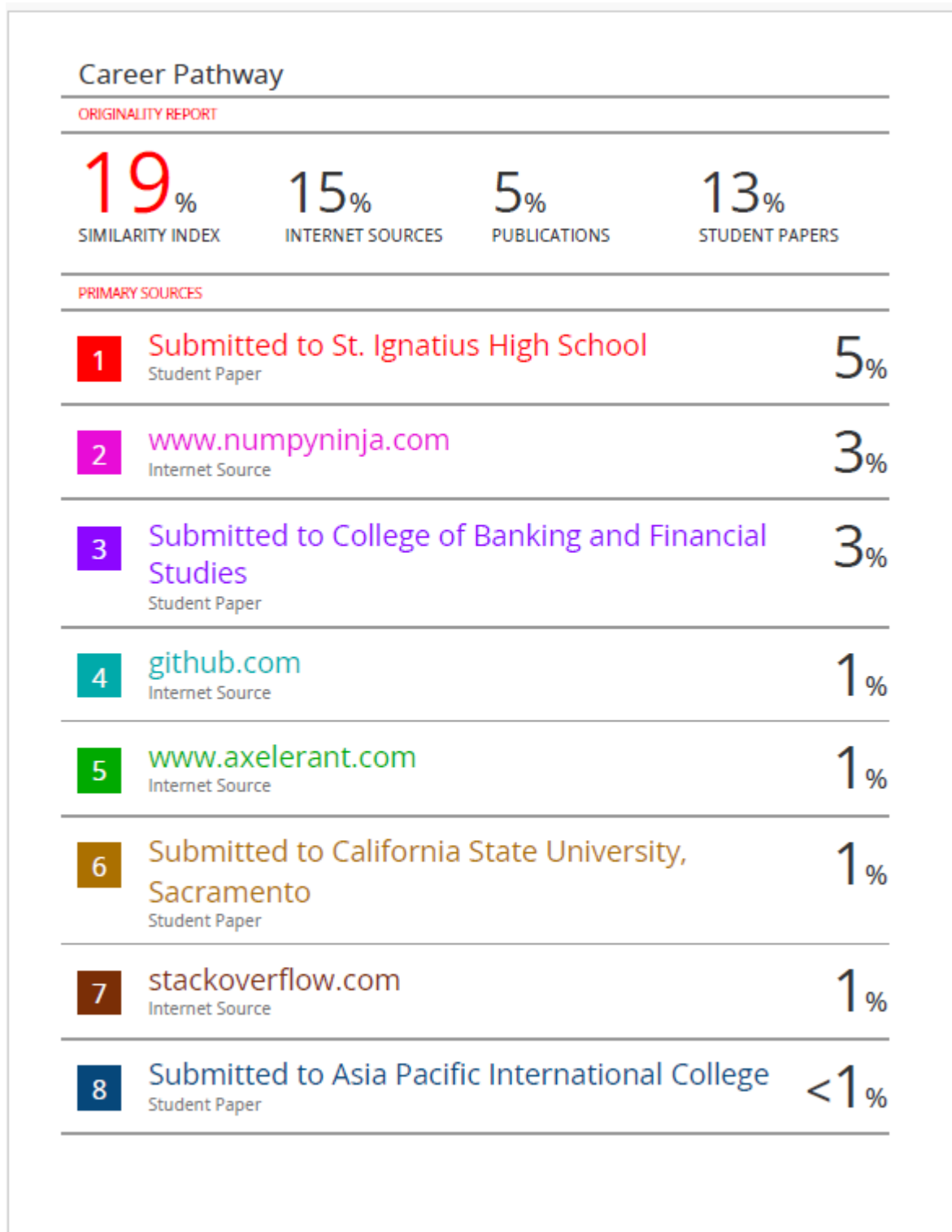
Fig 9.3.2



Fig 9.3.3**Fig 9.3.4**

Fig 9.3.5

9.5.1 PLAGIARISM



9	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1 %
10	pdfcoffee.com Internet Source	<1 %
11	core.ac.uk Internet Source	<1 %
12	ir.jkuat.ac.ke Internet Source	<1 %
13	www.coursehero.com Internet Source	<1 %
14	www.seleniumframework.com Internet Source	<1 %
15	Submitted to University of Calgary Student Paper	<1 %
16	code.google.com Internet Source	<1 %
17	Submitted to University of Westminster Student Paper	<1 %
18	Submitted to University of Newcastle Student Paper	<1 %
19	moldstud.com Internet Source	<1 %

20	Asaad, Omar. "Citizenpal", Universidade de Coimbra (Portugal), 2024 Publication	<1 %
21	Pankaj Jalote. "A Concise Introduction to Software Engineering", Springer Science and Business Media LLC, 2025 Publication	<1 %
22	Submitted to Gitam University Student Paper	<1 %
23	www.vskills.in Internet Source	<1 %
24	Submitted to Swinburne University of Technology Student Paper	<1 %
25	www.researchpublish.com Internet Source	<1 %
26	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
27	Blanco-Oliver Antonio, Lara-Rubio Juan, Irimia-Diéguez Ana, Liébana-Cabanillas Francisco. "Examining user behavior with machine learning for effective mobile peer-to-peer payment adoption", Financial Innovation, 2024 Publication	<1 %

28	Submitted to CSU, San Jose State University Student Paper	<1 %
29	Sufyan bin Uzayr. "Conquering JavaScript - The Practical Handbook", CRC Press, 2023 Publication	<1 %
30	Submitted to University of Warwick Student Paper	<1 %
31	Submitted to University of Wollongong Student Paper	<1 %
32	dsmworldian.blogspot.com Internet Source	<1 %
33	fdocuments.us Internet Source	<1 %
34	studentsrepo.um.edu.my Internet Source	<1 %
35	arkajainuniversity.ac.in Internet Source	<1 %
36	docshare.tips Internet Source	<1 %
37	dspace.chitkara.edu.in Internet Source	<1 %
38	riogrande.cs.tcu.edu Internet Source	<1 %
www.iare.ac.in		

39	Internet Source	<1 %
40	www.interviewbit.com Internet Source	<1 %
41	ir.juit.ac.in:8080 Internet Source	<1 %
42	resumecat.com Internet Source	<1 %
43	vdocuments.site Internet Source	<1 %
44	www.essaysauce.com Internet Source	<1 %
45	www.lambdatest.com Internet Source	<1 %
46	Guojun Gan. "Data Clustering in C++ - An Object-Oriented Approach", Chapman and Hall/CRC, 2019 Publication	<1 %
47	Lucjan Stapp, Adam Roman, Michaël Pilaeten. "ISTQB® Certified Tester Foundation Level", Springer Science and Business Media LLC, 2024 Publication	<1 %
48	Submitted to Sim University Student Paper	<1 %

49	Sujay Raghavendra. "Java Testing with Selenium", Springer Science and Business Media LLC, 2024 Publication	<1 %
50	eprints.upj.ac.id Internet Source	<1 %
51	repository.cuilahore.edu.pk Internet Source	<1 %
52	www.nakov.com Internet Source	<1 %
53	Zhimin Zhan. "Selenium WebDriver Recipes in C#", 'Springer Science and Business Media LLC', 2015 Internet Source	<1 %
<hr/>		
<div>Exclude quotesOff</div> <div>Exclude bibliographyOff</div> <div>Exclude matchesOff</div>		

9.5.2 PLAGIARISM AI REPORT



Page 1 of 94 - Cover Page

Submission ID trnoid::1-3193808295

Abhidarsh K S

Career Pathway



Quick Submit



Quick Submit



Amal Jyothi College of Engineering

Document Details

Submission ID

trnoid::1-3193808295

Submission Date

Mar 25, 2025, 10:10 AM GMT+5:30

Download Date

Mar 25, 2025, 10:15 AM GMT+5:30

File Name

MainProject_CareerPathway_4.docx

File Size

2.6 MB

92 Pages


13,471 Words

82,988 Characters



Page 1 of 94 - Cover Page

Submission ID trnoid::1-3193808295


Page 2 of 94 - AI Writing Overview
Submission ID tncoid::1:3193808295

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI-generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI-generated as AI-generated and AI-paraphrased or likely AI-generated and AI-paraphrased writing as only AI-generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.


AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).


The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.




Page 2 of 94 - AI Writing Overview
Submission ID tncoid::1:3193808295

Attach Plagiarism Report