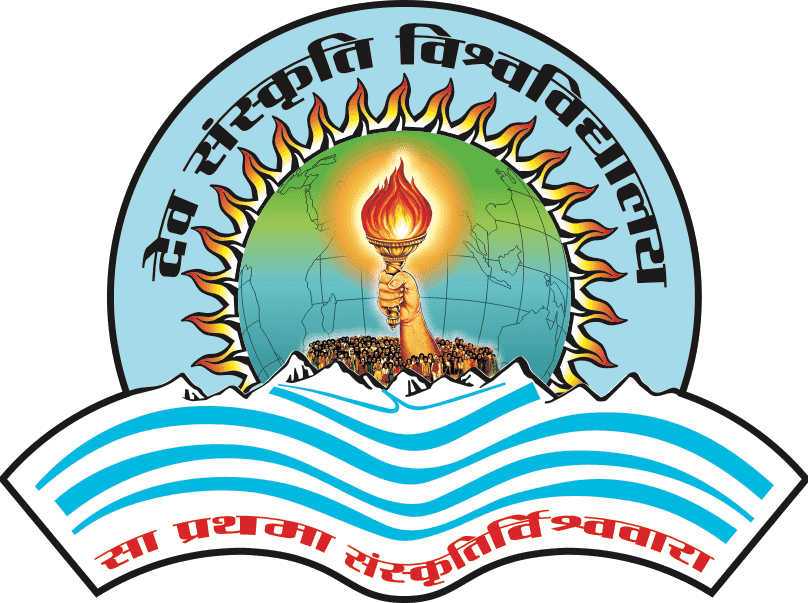
A Project Report on

**Skill Acquisition and Tracking System**

Submitted

in particular fulfilment of the requirements of the degree

**Bachelor of Computer Application**



**Academic Year**

2022-2023

**Submitted on**

06 / 05 / 2023

**Submitted by:**

**Kushal Thakur - 2024026**

**Kaivalya Sao – 2024004**

**Submitted To:**

**Dr. Naveen Kumar Pandey**

Department of Computer Science

**Department of**

**Dev Sanskrit Vishwavidyalaya, Gayatrikunj, Haridwar, Uttarakhand, 249411**

[**www.dsvv.ac.in**](http://www.dsvv.ac.in)

Index

Declaration

We, Kaivalya Sao & Kushal Thakur, students of Bachelor of Computer Applications (BCA) at Dev Sanskriti Vishwavidyalaya, hereby declare that the project entitled "Skill Tracking & Acquisition System" submitted for partial fulfillment of the requirement for the Bachelor of Computer Application degree is my original work and has not been submitted to any other institution for the award of any other degree or diploma.

The project involved the use of many new technologies such as D3.js and Firebase to develop a web-based platform that allows users to track their skills and acquire new ones. The project's objectives were to provide an intuitive user interface, accurate skill tracking and assessment, and personalized learning recommendations based on user preferences and performance.

The project is a result of extensive research, analysis, and experimentation carried out by me under the guidance of Dr. Naveen Kumar Pandey, a faculty member of the Department of Computer Science. I have made an honest effort to give due credit to all sources of information and data used in the project, and to ensure that the project is unique and innovative in its design and functionality.

I understand that any breach of this originality declaration may result in serious consequences, and I declare that the information presented in this project is true and accurate to the best of my knowledge.

Yours faithfully,

(Signatures)

2024004 : Kaivalya Sao

2024026 : Kushal Thakur

**Certificate**

This is to certify that the project entitled "Skill Acquisition and Tracking System" submitted by Kaivalya Sao and Kushal Thakur, students of Bachelor of Computer Applications (BCA) at Dev Sanskriti Vishwavidyalaya, is a record of original work carried out by the students under the guidance of "Dr. Naveen Kumar Pandey", a faculty member of the Department of Computer Science.

The project is submitted as a partial fulfillment of the requirement for the BCA degree and has not been submitted to any other institution for the award of any other degree or diploma.

The work embodied in this project is original and has been accomplished by the students through extensive research, analysis, and experimentation. The students have made an honest effort to give due credit to the sources of information and data used in the project.

This certificate is being issued in good faith and wishing students very good luck for their bright future.

Dr. Naveen Kumar Pandey

Assistant Professor

Department of Computer Science

Dev Sanskriti Vishwavidyalaya

Date: 5/5/2023

Acknowledgement

We would like to express our heartfelt gratitude to everyone who has contributed to the successful completion of my software development graduation project.

We are grateful to the Department of Computer Science at Dev Sanskriti Vishwavidyalaya for providing me with the necessary resources and infrastructure to undertake this project. We cannot thank enough the Head of the Department, Dr. Rajeshwari Trivedi and our project guide, Dr. Naveen Kumar Pandey, our batch coordinator Mr. Swapnil Gaidhani, for their valuable guidance, constructive feedback, and unwavering support throughout the project. We would also like to express our gratitude to Mr. Nischal Rai for their continued mentorship, it has been invaluable to us, and we are truly grateful for their efforts.

We would also like to take this opportunity to express our heartfelt thanks to our friends for their emotional support and motivation during the project. We are grateful to our families for being pillars of strength and encouragement throughout our academic journey.

Finally, we extend our appreciation to all the individuals and organizations who have contributed to the success of this project, including our academic advisors, instructors, and industry partners who have provided us with valuable feedback & technical expertise.

Once again, we express our sincere gratitude to everyone who has helped make this project a success.

Abstract

Introduction:

In today's fast-paced and ever-changing world, learning new skills is more important than ever. However, figuring out where to start and what to learn can be a daunting task. That's where our Skill Tracking & Acquisition System comes in. Our system is designed to help users efficiently and effectively learn new skills by providing a step-by-step path tailored to their current skill set.

Features:

* Skill Tracking: Users can track the skills they possess and visualize their journey throughout the lifetime of the application. This feature will help users understand their current skill set and identify areas for improvement.
* Skill Path Recommendation: Our system will analyze the user's current skill set and provide a personalized path to learn a new skill. The path will take into account the user's existing knowledge and suggest the most efficient way to learn the new skill.
* The Skill Tracking & Acquisition System will be built using a combination of HTML, CSS, and JavaScript to create an interactive user interface that is both visually appealing and functional. Firebase will be used as the backend database to store user data, including their current skills and progress on learning new skills.
* To visualize the skills and their relationships in a graph-like manner, the D3.js library will be utilized. D3.js is a popular JavaScript library that provides powerful data visualization capabilities, making it an ideal choice for representing complex relationships between skills in an easy-to-understand format.

Conclusion:

Our Skill Tracking & Acquisition System is an innovative solution to the problem of learning new skills. By providing a personalized path to learn new skills and visualizing the user's progress, our system will make the learning process more efficient, effective, and enjoyable. In the future we could implement machine learning algorithms that will suggest personalized learning paths based on a user's learning style and progress.

Additionally, the system could be expanded to include gamification elements, such as badges and rewards, to motivate users to continue learning and achieve their goals.

# **List Of Figures**

|  |  |  |
| --- | --- | --- |
| Figure Number | Title | Page Number |
| 6. A | Context Level DFD | 19 |
| 6. B | Level 1 DFD | 20 |
| 6. C | Level 2 DFD | 20 |
| 7. A | Data Model of Skills | 26 |
| 7. B | Data Model of Users | 26 |
| 8.A | Final Rendered Graph | 29 |

Chapter 1

## **Introduction:**

The problem that the Skill Tracking & Acquisition System aims to solve is the difficulty that people face in identifying where to start and how to efficiently and effectively learn new skills. With so much information available, it can be overwhelming to figure out the best path to take. The system aims to simplify this process by providing a personalized path based on the user's existing knowledge and skills, making it easier for them to acquire new skills and improve their abilities. Additionally, by tracking their progress and visualizing their journey, users can stay motivated and focused on their learning goals.

* The problem: Difficulty in identifying the best path to learn new skills.
* The solution: Personalized learning path based on existing knowledge.
* Context: Growing demand for efficient and effective ways to learn new skills.

Our software project aims to provide a solution to this problem by offering an efficient and effective way to learn new skills. The software application will provide a step-by-step path to learn a skill, taking into account the user's existing knowledge and providing the most efficient path to learn the skill. The application will also keep a record of the user's skills, allowing them to track their progress and visualize their journey.

## **System Context:**

The context for this project is the education industry, where there is a growing demand for efficient and effective ways to learn new skills.

* Our target audience is individuals who wish to learn new skills, ranging from students to working professionals.
* The Skill Tracking & Acquisition System is aimed at individuals of all ages and backgrounds who want to learn new skills.
* It may be for personal or professional growth. It has the potential to revolutionize the way people learn new skills and acquire knowledge.

## **Key Concepts:**

* The Skill Tracking & Acquisition System is a web-based application designed to help users learn new skills efficiently.
* The application guides users through a step-by-step path or paths to learn a skill, taking into account their current knowledge and skills.
* The application utilizes technologies such as HTML, CSS, and JavaScript for the front-end, Firebase for the back-end, and D3.js for data visualization.
* The system provides a graphical user interface that visualizes skills in a node-like structure, with fundamental skills and their related skills connected together.
* The interface helps users track their progress and visualize their journey in acquiring new skills.

Additionally, the application records the skills that a user currently possesses, allowing them to keep track of their progress and see areas where they need to improve.

In this document, we will provide an overview of the software application, its features, and the technologies used. Additionally, we will discuss the system's future scope and potential for expansion to cater to the needs of a wider audience.

**Chapter 2: Project Overview**

Objectives:

* To provide a clear path for users to learn new skills or achieve their desired career goals.
* To guide users through the learning process in the most efficient and effective way possible.
* To suggest the most effective resources or guides for learning a new skill, including academic and non-academic skills.
* To prioritize the most important skills to learn next based on the user's current skill set.
* To recommend fundamental skills that will benefit the user in the long run.
* To categorize and assign domains to groups of skills to help users understand the relationships between different skills.
* To provide a timeline view of the user's learning journey from start to current stage.
* To allow users to add details, subgroups, or granular information into another graph for further detail.
* To offer an interactive and visually appealing user interface to enhance the learning experience.

Scope of the Project:

The scope of the project involves creating a software application that enables users to learn new skills or achieve their desired career goals. This software will guide users along a clear path for acquiring new skills while taking into account their existing skill set. The software will categorize and assign domains to groups of skills to help users understand the relationships between different skills.

Furthermore, the software will prioritize the most important skills for users to learn next based on their current skill set and recommend fundamental skills that will benefit them in the long run. It will recommend relevant resources, including academic and non-academic materials, to aid users in their learning journey. Initially, the recommendations will be provided manually by the developers, but as the userbase grows, suggestions will be based on people who have already achieved similar goals, with the most effective path or guide selected through a rating system.

The software will offer an interactive and visually engaging user interface, featuring a graph-like representation of skills and nodes to aid users in understanding the relationships between skills. The software will also provide a timeline view of the user's learning journey, illustrating progress from the start to the current stage. Additionally, the software will allow users to add details, subgroups, or granular information into another graph for further detail.

The project's scope is limited to developing a system that recommends the best skills to learn next based on the user's current skills and goals. However, the scope of the project will not include any religious or political recommendations.

Impact of the Project:

The project will help and contribute to the users in several ways. Firstly, the software will enable users to learn new skills in a more efficient and effective way, taking into account their existing skill set. This will allow users to achieve their career goals and enhance their personal development, leading to a more fulfilling and successful life.

Secondly, the software will provide guidance on the fundamental skills that users should learn, which will benefit them in the long run. This will help users to plan their learning journey more effectively and gain a more comprehensive understanding of the relationships between different skills.

Thirdly, the software will provide an interactive and visually engaging user interface, making the learning experience more enjoyable and engaging for users. This will encourage users to continue learning and progressing, leading to greater satisfaction and motivation.

Overall, the project will help and contribute to the stakeholders and users by providing a valuable tool for learning and personal development, enhancing their skills and knowledge, and ultimately improving their quality of life.

# **Chapter 3: System Study**

Tracking the current skills that the user possesses is an important aspect of our Skill Tracking & Acquisition System. The system aims to simplify the process of learning new skills by providing a personalized path based on the user's existing knowledge and skills, making it easier for them to acquire new skills and improve their abilities.

To achieve this, the system will record the skills that a user currently possesses, allowing them to keep track of their progress and see areas where they need to improve. This feature will help users identify the gaps in their knowledge and develop a more targeted learning plan.

## **3.1 Existing Systems on Similar Scope**

There are several existing systems that aim to provide personalized learning paths and track progress towards learning new skills. These systems provide a platform for individuals to gain knowledge and enhance their abilities. But most of these have a standard approach for suggesting what paths the user should take.

* Like Skillshare - Skillshare is an online learning platform that offers thousands of classes and courses on a variety of topics, including art, design, business, technology, and more. Skillshare's primary focus is on creative skills, such as graphic design, photography, and writing.

These systems are standardized with little to no flexibility and none of them has the ability to track the current skills or skill levels that a user might possess. This can restrict growth opportunities. Only some applications are able to partially visualize and track the skills of a user.

These include mind mapping software like

* MindMeister
* XMind
* Coggle

These can help individuals create a visual representation of their skills and knowledge. Users can create nodes for different skills and subskills and link them together to create a skill tree and track their skills.

## **3.2 Drawbacks of Existing Systems and Limitations**

However, the existing tools are not specifically designed to track skill acquisition or provide personalized learning paths. While it can be used to represent skills and knowledge, it does not have the ability to suggest learning paths or provide insights into the user's progress towards a specific skill or goal. Other tools for creating visual representations of knowledge and skills may exist. However, they lack the personalized learning and progress tracking features that our proposed system aims to provide.

There are major improvements in our proposed Skill Tracking & Acquisition System as compared to the existing systems:

* Personalized Learning Path: Our system provides a personalized learning path based on the user's existing knowledge and skills, whereas existing systems may offer a standardized approach to learning.
* Skill Tracking: Our system places a strong emphasis on skill tracking, allowing users to keep track of their current skills, progress, and areas where they need improvement. Many existing systems do not provide this level of detail.
* Data Visualization: Our system utilizes data visualization tools, such as D3.js, to provide a graphical representation of the user's skill tree and progress. This makes it easier for users to see their progress and visualize their learning journey.
* Flexibility: Our system is flexible and can be tailored to meet the needs of a wide range of users, from students to working professionals. This level of flexibility is not always offered by existing systems, which may be targeted towards specific industries or skill sets.

Overall, our system aims to provide a more personalized, detailed, and flexible approach to skill acquisition and tracking, using cutting-edge technology to enhance the user experience.

# **Chapter 4: System Analysis**

System Analysis for Skill Tracking & Acquisition System

### **Problem Definition:**

* The current problem is that there is a lack of a personalized and flexible skill acquisition system that takes into account the user's current knowledge and skills.
* Most existing systems have a standard approach to suggesting learning paths and do not track the user's current skills. This can limit growth opportunities and hinder the user's ability to acquire new skills efficiently.
* The goal is to create a system that guides the user through a step-by-step path or paths to learn a skill, taking into account their current knowledge and skills, providing a flexible and personalized learning experience.

### Stakeholder Analysis:

Identifying the stakeholders of the system. For the particular system the stakeholders include

* The users of the application who want to learn new skills efficiently.
* The faculty members involved in the development process, their inputs and suggestions are valuable during development and implementation.
* The designers and developers of the project who are responsible for creating and maintaining the system.

Understanding the needs and requirements of each stakeholder group is crucial to the success of the system.

* The users would want a user-friendly interface that provides them with the right learning path that meets their goals.
* While us as the developers would want a system that is reliable, efficient, and provides value to the users.

### **Data collection for the System**

Following our initial analysis, we further gathered data on the platforms that provide similar functionality to that of the Skill Acquisition & Tracking System. Some of the highlighted information is given below.

Skillshare:

* Skillshare is an online learning platform that offers thousands of classes and courses on a variety of topics, including art, design, business, technology, and more.
* Users can browse and enroll in courses, and can also create their own courses to share their expertise with others.
* Skillshare has a standardized approach to course content and user progress tracking, with little flexibility for personalized learning paths or skill tracking.

Mind Mapping Software:

* MindMeister is a mind mapping software that allows users to create visual representations of their thoughts, ideas, and plans.
* Users can create nodes for different skills and subskills, and link them together to create a skill tree and track their skills.
* MindMeister has limited functionality for tracking skill levels or progress, and is primarily a tool for organizing and visualizing information.

### **Finalizing Requirements**

The Skill Tracking & Acquisition System offers a

* Personalized learning path based on the user's existing knowledge and skills
* Emphasizes skill tracking
* Utilizes data visualization tools to provide a graphical representation of the user's skill tree and progress
* Is flexible enough to meet the needs of a wide range of users.

This sets it apart from existing systems and by offering these features, our system aims to provide a unique and effective way for users to learn new skills and track their progress.

## **4.1 Purpose**

Our system aims to empower users in acquiring new skills efficiently and effectively. Through a personalized learning experience, we provide a flexible and tailored learning path based on their existing knowledge and skills. This ensures that users have a seamless transition into the learning process and are able to pick up new skills quickly.

One of the key features of our system is the ability to facilitate skill tracking and progress monitoring. Users can track their current skills and progress. This promotes continuous growth and improvement, as users are able to identify areas where they need to focus their attention.

## **4.2 Scope**

* Comprehensive Skill Tracking: Our system provides an in-depth analysis of the user's skill level, progress, and areas where they need improvement. This allows users to set realistic goals and track their progress over time.
* Real-Time Progress Updates: Our system updates in real-time, allowing users to see their progress as they learn new skills and complete tasks.
* Adaptive Learning: Our system uses the data collected through skill tracking to adapt the learning path and ensure that users are progressing at a pace that is right for them.
* Detailed Analytics: Our system provides detailed analytics on user performance, including proficiency levels, completion rates, and time spent on each skill. This helps users identify areas where they need to improve and make informed decisions about their learning path.
* Customized Learning Plans: Based on the user's current skill level and progress, our system generates a customized learning plan that helps them achieve their goals more efficiently.

## **4.3 Need for the proposed system**

The Skill Acquisition and Tracking System is needed to provide users with a personalized, flexible, and efficient learning experience. By utilizing technologies such as HTML, CSS, JavaScript, Firebase, and D3.js, the system can offer a tailored learning path based on the user's current skills and knowledge, as well as track their progress and provide data visualization for motivation and engagement. This sets it apart from existing systems, which may offer a standardized approach to learning and lack detailed skill tracking and progress monitoring. Ultimately, the system aims to empower users to acquire new skills and promote continuous growth and improvement.

## **4.4 Feasibility Study**

In the case of our proposed Skill Acquisition and Tracking System, a thorough feasibility study will help us assess if the project is viable and if it can be completed within the specified timeframe and budget. The following sections provide a detailed analysis of each feasibility factor.

**Technical Feasibility:**

* The proposed system requires the use of modern web development technologies such as HTML, CSS, JavaScript, and Firebase for back-end development, which are widely available and easily accessible. Therefore, the technical feasibility of the system is high.
* Since the system is web-based, it can be accessed from any device with a web browser and an internet connection, regardless of the operating system or hardware platform.
* Web-based systems can be easily scaled to accommodate large numbers of users and increased traffic.
* The use of D3.js for data visualization is also feasible, as it is a widely used and well-documented tool for creating interactive visualizations.

**Operational Feasibility:**

* The proposed system will require the availability of a stable and reliable internet connection, as it is a web-based application. Therefore, the system's operational feasibility will depend on the users' access to a stable internet connection.
* The system's usability will be critical for its operational feasibility, as it must be intuitive and easy to use for users to fully benefit from its features. Extensive user testing and feedback will be necessary to ensure the system's usability.

**Economic Feasibility:**

* The cost of developing and maintaining the system will be a crucial factor in its economic feasibility. The use of open-source technologies and frameworks will significantly reduce development costs, while cloud hosting services can provide cost-effective hosting and maintenance solutions.

Based on this feasibility analysis, the Skill Acquisition and Tracking System has a high potential for success and is feasible to implement.

# Chapter 5: SRS Document

This section outlines the software requirements specification (SRS) for the proposed Skill Acquisition and Tracking System. The SRS serves as the official statement of what is required by the system developers and includes both the requirements definition and specification.

## **5.1 Functional Requirements**

1. The system shall allow users to create an account and log in to their personalized dashboard.
2. The system shall create a personalized learning path for the user based on their pre-test results and provide them with learning materials and resources.
3. The system shall track the user's progress as they complete learning activities and provide feedback on their performance.
4. The system shall allow users to track their skill acquisition over time through interactive data visualization tools.
5. The system shall allow users to set learning goals and track their progress towards achieving those goals.

## **5.2 Non-Functional Requirements**

1. Usability: The system shall be easy to use and navigate for users of all skill levels.
2. Performance: The system shall respond to user requests in a timely manner and provide a smooth user experience.
3. Scalability: The system shall be able to handle increased user demand and workload without significant degradation in performance.
4. Availability: The system shall be operational and accessible to users at all times, with minimal downtime for maintenance and updates.
5. Security: The system shall have measures in place to protect user data and prevent unauthorized access or manipulation.
6. Compatibility: The system shall work with a variety of web browsers and operating systems, and be mobile responsive.
7. Maintainability: The system shall be easy to modify and maintain over time, with clear documentation and organized code.
8. Reliability: The system shall perform consistently and without failure over a long period of time.
9. Portability: The system shall be easily deployable to different environments or platforms.

## **5.3 Hardware Requirements**

To operate a web application on the user end, the following hardware is typically required:

* A device: This can be a desktop computer, laptop, tablet, or smartphone.
* Operating System: The device should have an operating system installed such as Windows, macOS, iOS, Android, or Linux.
* Web Browser: The user needs a web browser to access the web application. Popular web browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari, and Opera.
* Internet Connection: An internet connection is required to access the web application. A high-speed internet connection is recommended for optimal performance.

These hardware requirements are minimal and can be easily met by most modern devices.

## **5.4 Software Requirements**

To run our application, a user needs a web browser which can interpret and render the HTML, CSS, and JavaScript code that makes up the web application.

* Operating System: Windows 7, Windows 8, Windows 10, macOS, or Linux.
* Web Browser: Google Chrome, Mozilla Firefox, Apple Safari, or Microsoft Edge.
* Accounts: A Google account to login to the web app.

# **Chapter 6: System Design Specification**

# **6.1 Architecture**

* The data architecture involves using Firebase as the primary database for the application.
* User authentication and authorization is handled by Firebase Authentication.
* The user's data is stored in the Firestore database, which is a NoSQL document database.
* The data is structured in collections and documents, where each collection represents a category of data, and each document represents a record within that category.
* The application uses Firebase Cloud Functions to process and manipulate data on the server-side.
* The data is accessed and manipulated on the client-side using the Firebase SDK, which provides a set of APIs for querying and modifying data in real-time.

# **6.2 Data Flow Diagrams**

DFDs will specify the flow of data and processes within the system. They will outline the inputs, outputs, and processes involved in the various modules of the system such as

* Authentication,
* Adding skills, and
* Displaying the Master graph & the Skill Graph

By visualizing the system's processes and data flow, DFDs can help identify potential bottlenecks, improve system efficiency, and ensure that the system meets its functional requirements.

We will be providing multiple levels of Data Flow Diagrams (DFDs) that can help in understanding the system at different levels of detail.

* Context level DFD: This will show a high-level overview of the system and its interactions with external entities. It will include the main processes and data flows of the system.
* Level 1 DFD: This will provide a more detailed view of the system, showing the processes and data flows at a lower level. It will also show the interactions between the processes and data stores.
* Level 2 DFD: This will show an even more detailed view of the system, breaking down the processes and data flows into smaller, more manageable components. It will include additional details on the data being processed and stored within the system.

Each level of DFD will provide a deeper understanding of the system's processes and data flows, making it easier to identify areas for improvement and optimization.

### **6.2.1 Context Level DFD**

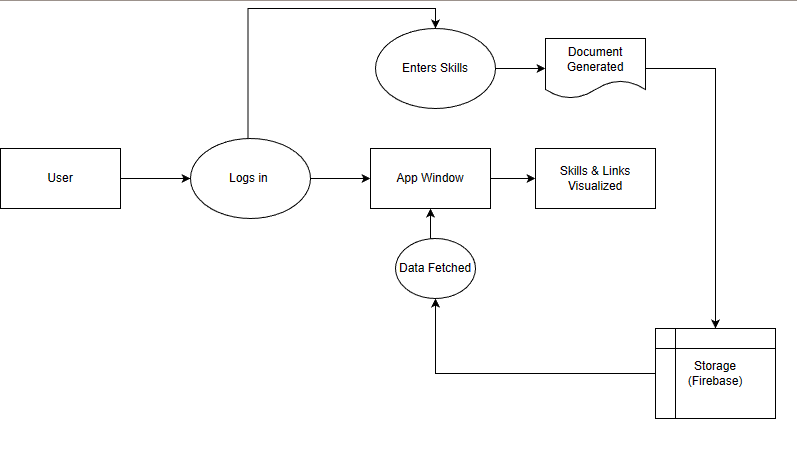


Figure 6. A Context Level DFD

### **6.2.2 Level 1 DFD**

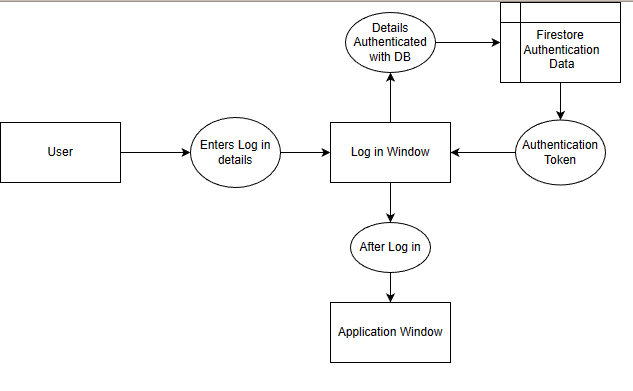


Figure 6. B Level 1 DFD

### **6.2.2 Level 2 DFD**

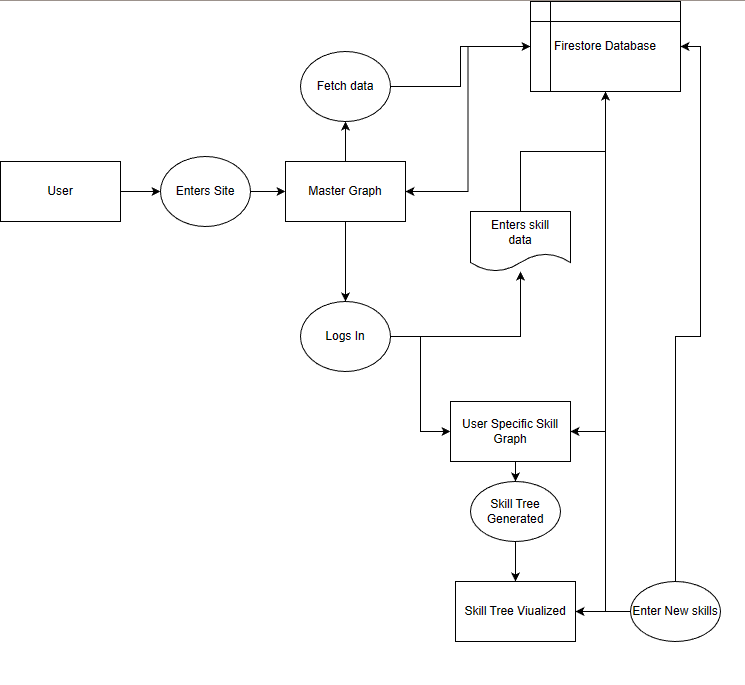


Figure 6. C Level 2 DFD

## **6.3 Use Cases**

Use cases are descriptions of specific scenarios in which a user interacts with the system to achieve a particular goal. They provide a clear and concise picture of the functional requirements of the system, ensuring that it meets the needs of the end-users. The use cases for our project are mentioned below.

* + 1. **Login/Authentication Use Case:**
* Actor: User
* Preconditions: User has an account in the system and has not already logged in.
* Flow of events:
  + User enters their email and password.
  + System validates the credentials and grants access to the user.
  + User is directed to the main page.
* Post-conditions: User is logged in and has access to the system's features.
  + 1. **Add Skill Use Case:**
* Actor: User
* Preconditions: User is logged in and wants to add a new skill to their profile.
* Flow of events:
  + User clicks on the "Add Skill" button.
  + System prompts user to enter the name and prerequisite skills for the new skill.
  + User enters the required information and submits the form.
  + System validates the information and adds the new skill to the user's profile.
  + System updates the main graph to include the new skill.
* Post-conditions: User has successfully added a new skill to their profile and it is displayed on the main graph.
  + 1. **View Skill Progress Use Case:**
* Actor: User
* Preconditions: User is logged in and wants to track their progress in learning a particular skill.
* Flow of events:
  + User clicks on the skill they want to track on the main graph.
  + System displays a pop-up window showing the user's current proficiency level, completion rate, and time spent on the skill.
* Post-conditions: User has viewed their progress on a specific skill.
  + 1. **Generate Customized Learning Plan Use Case:**
* Actor: System
* Preconditions: System has access to the user's skill data and progress information.
* Flow of events:
  + System analyzes the user's current skill level, progress, and goals.
  + System generates a customized learning plan for the user based on the analysis.
  + System displays the learning plan to the user.
* Post-conditions: User has received a customized learning plan that helps them achieve their goals more efficiently.
  + 1. **Edit Skill Use Case:**
* Actor: User
* Preconditions: User is logged in and wants to edit a skill on their profile.
* Flow of events:
  + User clicks on the skill they want to edit on their profile.
  + System displays a form with the current information of the skill.
  + User updates the required information and submits the form.
  + System validates the information and updates the skill on the user's profile.
  + System updates the main graph to reflect the changes.
* Post-conditions: User has successfully edited a skill on their profile and it is displayed accurately on the main graph.

## **Technical Background**

Technical background on the technologies and tools used in the Skill Acquisition and Tracking System:

**Front-End:**

* HTML, CSS, and JavaScript: Used for building the front-end of the web application.
* D3.js: A JavaScript library used for data visualization and creating the force-directed graph.
* Font Awesome: A popular CSS framework used for icon in the web design.

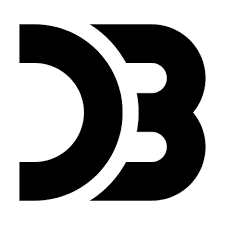
**Back-End:**

* Firebase: A cloud-based platform provided by Google used for serverless back-end development. Firebase provides several services like authentication, real-time database, cloud functions, and hosting.
* In this project, we use Firebase Authentication and Firestore Database.

**Other Tools:**

* Git: A version control system used for source code management and collaboration.
* GitHub: A web-based platform used for hosting and sharing code repositories.
* Visual Studio Code: An integrated development environment (IDE) used for writing and debugging code.
* Google Drive: To integrate all file and backup essential documents easily.

These technologies and tools were chosen for their efficiency, scalability, and ease of use. Firebase and D3.js were particularly useful in providing real-time updates and data visualization capabilities for the project.



# **Chapter 7: Database Design**

In our project, we will be using a database to store user data, such as user authentication information, user profiles, and skill data. We will be using Firebase Firestore as our database solution.

For the database structure, we will have three main collections: users, skills, and user\_skills. The users collection will store user authentication information and profile data, such as name, email, and profile picture. The skills collection will store skill data, such as skill name, prerequisite skills, and related skills. The user\_skills collection will store the skills that each user has added to their profile, along with any progress or notes related to that skill, this is going to be a referential collection based on the main skills collection.

To represent these entities and their relationships, we can create an Entity-Relationship (ER) diagram. The diagram would show the relationships between the three collections and the attributes associated with each entity. We can also create table structures to define the specific fields, data types, and constraints for each entity in the database.

In addition to the database structure, we will also have data structures for our application code. For example, we may use arrays or objects to store and manipulate skill data in our front-end code. We may also use data structures to store user authentication tokens and other session-related data.

## **7.1 Table Structures**

In our project, we are using Firebase Firestore as our database. Firestore is a NoSQL document-oriented database that stores data in collections and documents. Here is the table structure for each collection in our database:

1. **Users Collection:**

* uid (string): unique identifier for the user
* name (string): name of the user
* email (string): email address of the user
* skills (array of objects): array of skill objects that the user has added to their profile, each object containing the following fields:
  + skill\_id (string): unique identifier for the skill
  + skill\_name (string): name of the skill
  + skill\_description (string): description of the skill
  + skill\_level (number): level of the skill (1-5)
  + prerequisite\_skills (array of strings): array of skill IDs for the prerequisite skills for the current skill

1. **Skills Collection:**

* skill\_id (string): unique identifier for the skill
* skill\_name (string): name of the skill
* skill\_description (string): description of the skill
* prerequisite\_skills (array of strings): array of skill IDs for the prerequisite skills for the current skill

1. **UserSkills Collection:**

* user\_id (string): unique identifier for the user
* skill\_id (string): unique identifier for the skill
* skill\_level (number): level of the skill for the user (1-5)

These table structures allow us to store user information, skill information, and the relationship between users and skills in a structured and efficient way.

## **7.2 Data Structures**

Data structures refer to the way data is organized and stored in a computer program. In our project, some of the data structures used include:

* Arrays: Arrays are used to store and manipulate lists of values, such as the skills and prerequisite skills entered by the user. They allow for efficient access and manipulation of data.
* Objects: Objects are used to represent entities in the system, such as the user and the skills. They contain properties that define the characteristics of the entity and methods that allow for manipulation of the entity.
* Map: Used to associate keys with values, such as the relationship between a skill and its prerequisites or a skill and its level of proficiency.
* Set: Used to store a collection of unique elements, such as a set of skills that a user has.
* Graphs: Graphs are used to represent complex relationships between entities, such as the connections between different skills. They allow for visualization and analysis of the relationships between entities.

These data structures are used in conjunction with programming languages and frameworks such as JavaScript and D3.js to implement the functionality of the system.

## **7.3 Data Models**

Firebase as your database, which is a NoSQL cloud-based database. NoSQL databases do not follow the traditional relational model of databases and use various data models such as document-based, key-value, graph, and columnar. In the case of Firebase, it uses a document-based data model, where data is stored as JSON documents.

In this document-based data model, data is organized into collections, which can contain multiple documents.

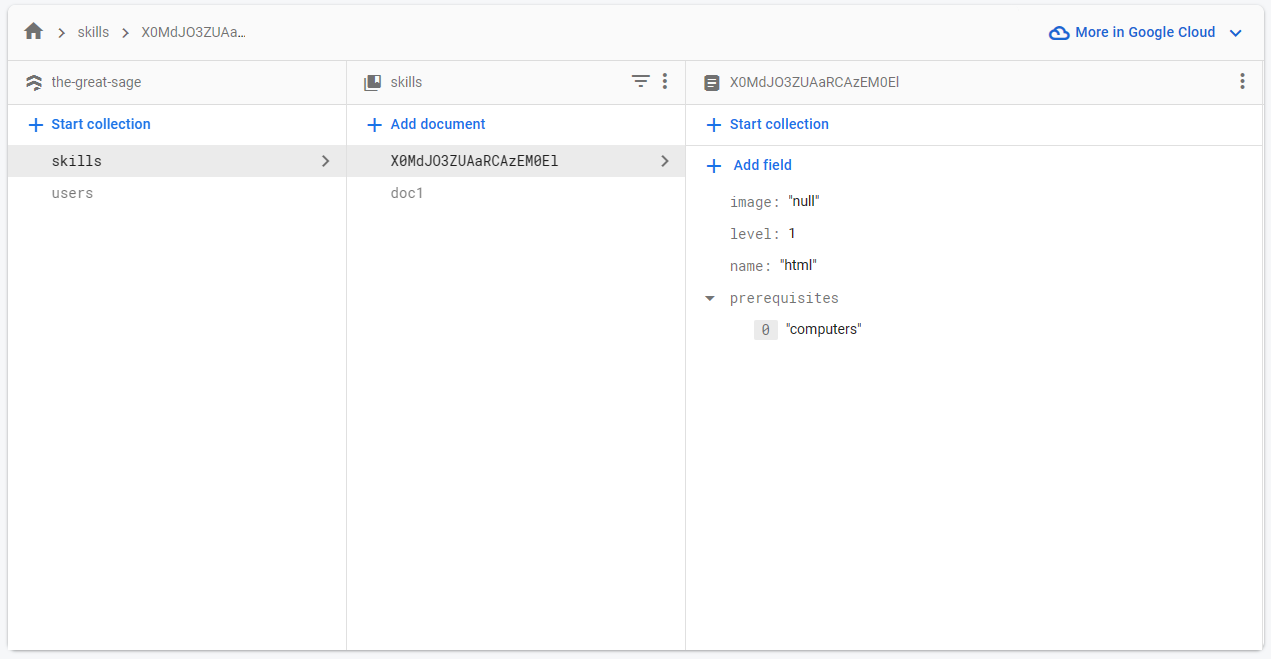


Figure 7. A Data Model Skills

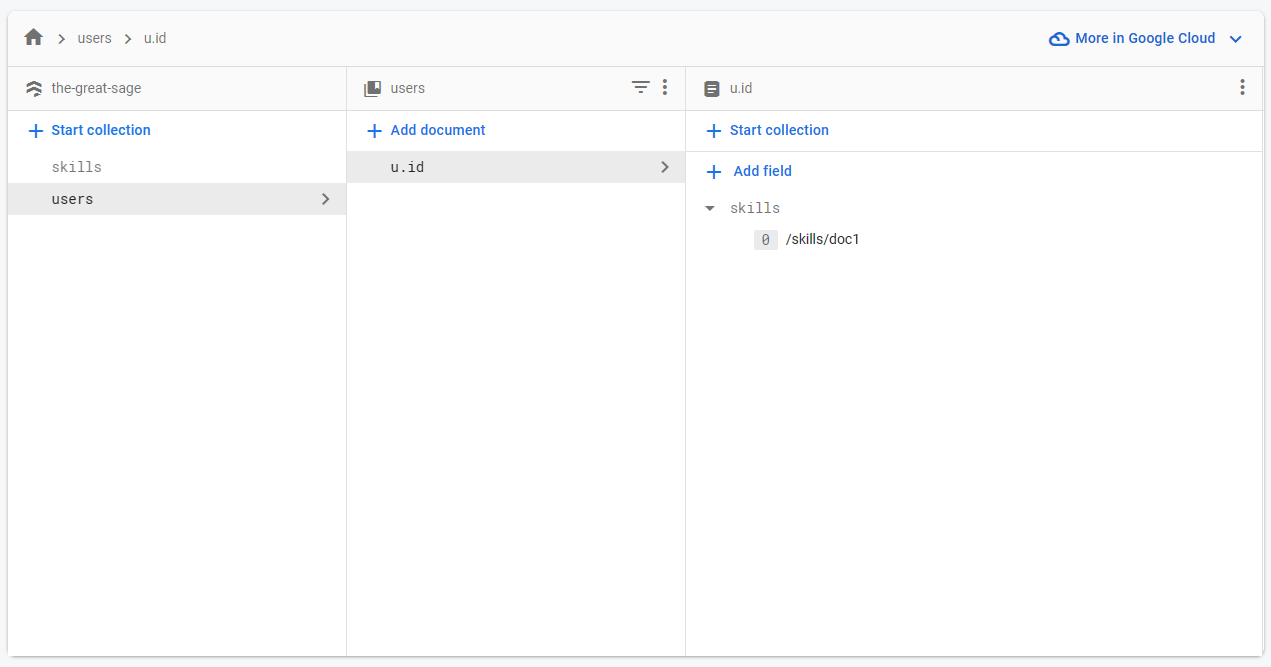


Figure 7. B Data Model Users

**Chapter 8: System Development**

This project aims to develop a knowledge and skill management system that provides users with a step-by-step path to learn new skills. The system will suggest the most efficient path to learn the skill in the most effective way possible. It will visualize the user's skills and their connections in a graph-like manner, allowing the user to easily identify the best skills to learn next. Additionally, the system will provide recommendations for resources to acquire knowledge in various domains. Overall, this system aims to help users manage their knowledge and skills effectively and efficiently.

**Design and Development Process :**

Our development process for this project involved starting with the front-end development and then designing the data structure with Firebase as the database and authentication provider. We initially considered using a Graph database to store the skills but ultimately settled on a table-like approach as it aligned better with our front-end and proved to be more feasible.

As for our decision-making process, we took a collaborative approach where we discussed and weighed different options for the project and made decisions based on what would best serve the needs of the users. We also took into consideration factors such as feasibility, scalability, and ease of implementation. Throughout the development process, we made sure to continuously communicate and collaborate to ensure that the project was progressing smoothly and meeting the requirements of the users.

**Key Features of the Software :**

The software is designed to help users visualize and track their skills and expertise. Some of the key features of the software include:

1. Graph Visualization : Instead of a traditional list or table format, we've created a visual graph that displays each skill and its relationship to other skills in a more intuitive way. This not only makes it easier for users to understand their own skills and how they relate to each other, but it also provides a more engaging and interactive user experience.
2. Skill tracking : The software allows users to input their skills and expertise, which are then displayed in a visual format. The user can see their skills represented as nodes in a graph, with links between related skills. This makes it easy to see which skills are related and how they fit together.
3. Personalized skill assessment: The software uses an algorithm to analyse a user's skills and provide a personalized assessment of their strengths and weaknesses. This helps users identify areas where they need to improve and focus their efforts on skill development.
4. Learning resources: The software provides access to a library of learning resources, including articles, videos, and tutorials. Users can search for resources based on their interests and skill level.

Overall, the software offers a unique combination of skill tracking, personalized assessment, collaboration, and learning resources. This makes it an innovative tool for anyone looking to develop their skills and expertise.

**Challenges and how we Overcame them :**

During the development process, we encountered several challenges. One of the major challenges was deciding on the appropriate data structure to use for storing skills. We initially considered using a Graph database, but we found it difficult to scale and incorporate with our front-end design. To overcome this, we decided to store skills in a table-like format, which was more compatible with our front-end design and easier to scale.

Another challenge we faced was implementing authentication and authorization for users. We overcame this by using Firebase as our backend, which provided easy-to-use authentication and authorization functionalities.

We also faced challenges with integrating the D3 graph library into our project, as it required a different data structure than what we were initially using. To overcome this, we had to modify our data structure and restructure our code to be compatible with the D3 library.

Finally, we encountered challenges with optimizing the performance of our application, particularly with regards to loading time and scalability. We addressed this by implementing caching techniques, optimizing our queries, and optimizing our database structure.

**Some unique code snippets of our project & Technical Documentation:**

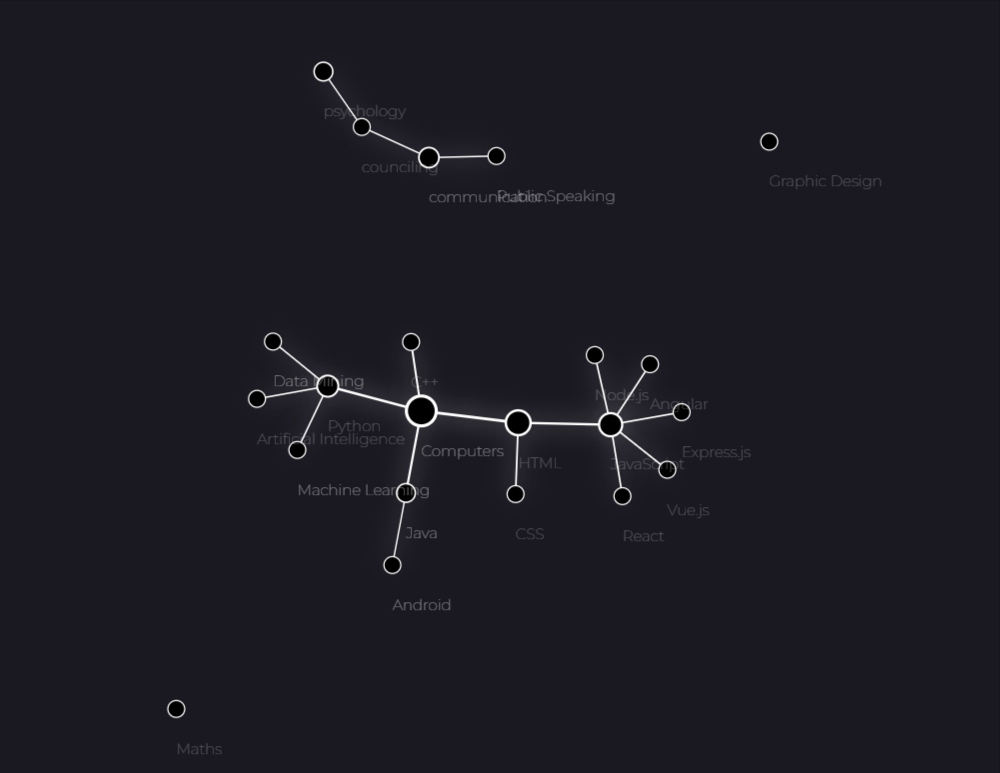
****

Figure 8. A: Final Rendered Graph

**1. Setting up the Simulation :**

This function sets up the force simulation for the graph and defines the behavior of the nodes and links in the graph. Here's a summary of what the function does:

1. Calls the generateLinks() and calculateWeight() functions to generate links between the skills and calculate the weights of the nodes.
2. Sets up the force simulation using the d3.forceSimulation() function and adds three forces: link, center, and charge.
3. Defines the colors for the skills and links using the d3.scaleOrdinal() function.
4. Sets up the node radius scale using the d3.scaleLinear() function.
5. Creates the links using the svg.append() and data().enter().append() functions, and sets their attributes such as stroke-width and stroke-opacity.
6. Creates the nodes using the same svg.append() and data().enter().append() functions, and sets their attributes such as radius, fill, stroke, and stroke-width. It also adds drag and click events to the nodes.
7. Adds labels to the nodes using the svg.append() and data().enter().append() functions, and sets their attributes such as font-size, opacity, and text.
8. Starts the simulation using the simulation.nodes() and simulation.force().links() functions and calls the ticked() function on every tick of the simulation.

­

function setupSimulation() {

*// Set up the force simulation*

*//calculating weight of all the skills*

  generateLinks()

  calculateWeight()

  let simulation = d3

    .forceSimulation()

    .force(

      'link',

      d3

        .forceLink()

        .distance(50)

        .id((*d*) => *d*.id),

    )

    .force('center', d3.forceCenter(width / 2, height / 2))

    .force(

      'charge',

      d3.forceManyBody().strength(-100).distanceMin(10).distanceMax(200),

    )

    .force(

      'collision',

      d3.forceCollide().radius((*d*) => Math.sqrt(2) \* 10),

    )

*//.force('radial',d3.forceRadial(150,width/2,height/2).strength(0.5))*

  addEventListener('resize', (*event*) => {

    width = +svg.node().getBoundingClientRect().width

    height = +svg.node().getBoundingClientRect().height

    simulation.force('center', d3.forceCenter(width / 2, height / 2)).restart()

  })

*// Define the colors for the skills and links*

  const color = d3.scaleOrdinal(d3.schemeCategory10)

*// Set up the node radius scale*

  const radius = d3

    .scaleLinear()

    .domain(d3.extent(data.skills, (*d*) => 0.5))

    .range([5, 30])

*// Create the links*

  link = svg

    .append('g')

    .attr('class', 'links')

    .selectAll('line')

    .data(data.links)

    .enter()

    .append('line')

    .attr('stroke-width', (*d*) => radiusValue(*d*))

    .attr('stroke', 'white')

    .attr('stroke-opacity', 1)

    .attr('class', 'link')

*// Create the skills*

  node = svg

    .append('g')

    .attr('class', 'skills')

    .selectAll('circle')

    .data(data.skills)

    .enter()

    .append('circle')

    .attr('r', (*d*) => `${Math.sqrt(*d*.weight \* 3) + 5 || 10}px`)

    .attr('fill', 'black')

    .call(drag(simulation))

    .on('click', (*d*) => showLabel(*d*))

    .on('mouseover', (*d*) => nodeHover(*d*))

    .on('mouseout', (*d*) => nodeUnhover(*d*))

    .attr('stroke', 'white')

    .attr('stroke-width', (*d*) => `${Math.cbrt(*d*.weight) || 10}px`)

*// Add labels to the skills*

  label = svg

    .append('g')

    .attr('class', 'label')

    .selectAll('text')

    .data(data.skills)

    .enter()

    .append('text')

    .attr('dy', `3em`)

    .attr('font-size', '12px')

    .style('font-weight', '100')

    .style('opacity', '0.3')

    .text((*d*) => *d*.id)

  node.append('title').text((*d*) => *d*.id)

  label.attr('dx', function (*d*) {

    const radiusValue = radius(*d*.count)

*return* radiusValue ? radiusValue + 5 : 0

  })

*// Start the simulation*

  simulation.nodes(data.skills).on('tick', ticked)

  simulation.force('link').links(data.links)

}

**2. Generating links between nodes :**

This function generates links between skills based on their prerequisites. It loops through all the skills in the data and for each skill, it checks its prerequisites. If a prerequisite exists, it creates a link between the prerequisite and the skill. The generated links are stored in the data.links array. This function is crucial for rendering the skill graph, as it provides the necessary connections between skills.

function generateLinks() {

  let generatedLinks = []

  data.skills.forEach((*skill*) => {

*skill*.prerequisites.forEach((*prerequisite*) => {

      generatedLinks.push({ source: *prerequisite*, target: *skill*.id })

    })

  })

  data.links = generatedLinks

}

**3. Authentication Module :**

This code imports Firebase Authentication modules, initializes authentication and Google auth provider, defines a function to handle Google sign in, and sets up a listener to check if a user is currently signed in. The signIn() function uses signInWithPopup to authenticate a user with Google and retrieve their credentials and user information. The onAuthStateChanged function listens for changes in the user's authentication state and logs their user object if they are authenticated. The auth and signIn functions are exported for use in other modules.

*import* { getAuth, onAuthStateChanged, GoogleAuthProvider, signInWithPopup, } *from* "https://www.gstatic.com/firebasejs/9.21.0/firebase-auth.js";

*import* {app} *from* "./firebase.js"

const auth = getAuth(app)

const provider = new GoogleAuthProvider();

function signIn() {

  signInWithPopup(auth, provider)

    .then((*result*) => {

*// This gives you a Google Access Token. You can use it to access the Google API.*

      const credential = GoogleAuthProvider.credentialFromResult(*result*);

      const token = credential.accessToken;

*// The signed-in user info.*

      const user = *result*.user;

*// IdP data available using getAdditionalUserInfo(result)*

*// ...*

    }).catch((*error*) => {

*// Handle Errors here.*

      const errorCode = *error*.code;

      const errorMessage = *error*.message;

      console.log(errorMessage)

*// The email of the user's account used.*

      const email = *error*.customData.email;

*// The AuthCredential type that was used.*

      const credential = GoogleAuthProvider.credentialFromError(*error*);

*// ...*

    });

}

onAuthStateChanged(auth, (*user*) => {

*if* (*user*) {

    console.log(*user*)

  } *else* {

  }

});

*export* {auth,onAuthStateChanged,signIn}

**Chapter 9: Testing**

**Test Goals for the Project :**

The testing goals for this particular project include:

1. To ensure that all components of the system are functioning as intended, such as the sign-in function, generate graph function, and other features.
2. To test the integration of all the components to ensure they work seamlessly together as a system.
3. To test the overall functionality of the system, including its ability to generate graphs, visualize data, and provide insights.
4. To verify that the system meets all requirements and specifications provided by the client.
5. To evaluate the performance of the system, including its speed, efficiency, and ability to handle large datasets.
6. To identify and fix any bugs, errors, or issues that may arise during testing, and to ensure that the system is robust and reliable.
7. To ensure that the system is user-friendly and easy to use, with clear instructions and guidance provided to users.
8. To document all testing processes and results, including any issues or bugs that were identified and fixed, and to provide feedback to the development team for further improvements.

**Types of Testes & How we did the Testing:**

1. Component Testing:
   * We tested each component of the system separately.
   * For example, we tested the signIn function and the generateGraph function.
   * We used unit testing frameworks like Jest to test these components.
2. Integration Testing:
   * Once we had tested all the components, we integrated them and tested them together.
   * For example, we tested the signIn function and the generateGraph function together.
   * We used integration testing frameworks like Cypress to test these components together.
3. System Testing:
   * Once we had tested the components together, we tested the overall functionality of the system.
   * For example, we tested if the user could sign in and generate graphs successfully.
   * We used end-to-end testing frameworks like Selenium to test the system.
4. Performance Testing:
   * We also ensured that the performance of the system was optimal.
   * For example, we tested how long it took for the system to generate a graph.
   * We used performance testing frameworks like Apache JMeter to test the system's performance.

Overall, we used a combination of unit testing, integration testing, end-to-end testing, and performance testing to ensure that our system was functional and performant.

**Conclusion :**

In summary, we conducted a comprehensive testing process for our software project that aimed to ensure its functionality, reliability, and performance. We started by setting clear testing goals that aligned with our project requirements and specifications. We then categorized our testing into unit testing, integration testing, system testing, and performance testing, and performed each step systematically. We started by testing each component separately, such as the sign-in function and the graph generation function, and then integrated them to test the overall functionality of the system. We also paid attention to performance testing and tested the software on different systems. We learned that testing is a crucial step in software development and helps to identify and fix bugs and issues before the product is released. By following a well-planned and structured testing process, we were able to ensure that our software met our requirements and was reliable, functional, and performant.

**Chapter 10: Implementation**

**The implementation process for this project:**

1. **Planning and design:** In this stage, we worked on defining the requirements and functionalities of the system. We also designed the UI/UX and finalized the technologies to be used.
2. **Development:** In this stage, we started building the software using the technologies that were finalized in the planning stage. We divided the development process into small tasks and worked on them in iterations, following an agile methodology. We also performed unit testing on each component as it was completed.
3. **Integration and testing:** After completing the development of each component, we integrated them and tested the system as a whole. We also tested the performance of the system to ensure that it would function optimally when used by multiple users.
4. **Deployment:** After testing and making necessary modifications, we deployed the software on a cloud platform (Firebase) to make it accessible to users. We also ensured that the software was secure and reliable before making it publicly available.
5. **Maintenance:** Once the software was deployed, we continued to maintain it by fixing any bugs that were caught and adding new features to improve its functionality.

Throughout the implementation process, we made sure that their requirements were met. We also followed industry best practices and standards to ensure the quality of the software.

Implementation process of our project involved several steps. Firstly, we had to understand the requirements of the project and identify the necessary technologies to use. We then went through the learning process of the technologies we did not know before, such as D3 and Firebase Hosting. After acquiring the necessary knowledge, we started the implementation process by creating a basic HTML, CSS, and JavaScript structure for the project. We then developed the main functionality of the software, which included creating functions for data input, visualization, and authentication. After implementing the core functionalities, we proceeded to test the software thoroughly by categorizing our testing into different stages. We tested individual components, the integration of components, the overall functionality of the system, and the performance of the software. Finally, after successful testing, we hosted the software on Firebase to serve our users.

**Chapter 11: Conclusion & Future Scope**

**Future Scope:**

Future scope refers to potential areas of development or improvement that can be explored in the future to enhance the functionality or capabilities of a system. In the context of our project, some of future scopes are:

1. Integration with LinkedIn: In the future, we could integrate the software with LinkedIn to enable users to import their profile data directly into the system. This would save time and effort and help users get started more quickly.
2. Gamification: We could add gamification elements to the software to make the experience more engaging and fun. For example, users could earn points for completing tasks, levelling up as they progress, and unlocking rewards for achieving milestones.
3. Social Media Integration: We could also add social media integration to the software to enable users to share their skill graph on their social media profiles. This would allow them to showcase their skills and accomplishments to a wider audience and potentially attract more job opportunities.
4. Sharing Skill Graph as a Mini Resume: Users could also have the option to share a link to their skill graph as a mini resume to show off their skills to potential employers. This would be an innovative way for job seekers to differentiate themselves from others and stand out in a competitive job market.
5. User Feedback and Analytics: We could implement user feedback and analytics to gather insights into how users are interacting with the software, what features they find most useful, and what areas need improvement. This would help us to make data-driven decisions about how to improve the software over time.
6. Mobile Application: To increase accessibility, we could develop a mobile application for the software. This would enable users to access their skill graph and perform other functions on the go, making it more convenient and user-friendly.

**Conclusion:**

This project was an ambitious undertaking to build a web-based dashboard that could display visualizations of a user's Skills and help them Acquire more. We used a combination of front-end technologies like HTML, CSS, JavaScript, and D3.js, along with back-end technologies like Firebase to store user data and provide authentication services. The project was an excellent opportunity for us to learn new skills and technologies and apply them to a real-world problem.

Throughout the development process, we had to tackle various challenges, including learning new technologies, designing and implementing complex data visualizations, and integrating different components of the system. We also had to make sure that the application was easy to use, performant, and reliable.

Our testing process was thorough and helped us catch numerous bugs and issues that would have otherwise gone unnoticed. We tested each component of the system individually and then tested them together to ensure that they worked seamlessly. We also tested the performance of the system and optimized it wherever possible.

Overall, this project was an incredible learning experience, and we are proud of what we were able to accomplish. It taught us the importance of planning, communication, and collaboration in software development. We hope that our project can inspire others to take on ambitious projects, learn new technologies, and push their boundaries.

**Bibliography**

Here is the bibliography for the references used in the development of our project. These sources provided valuable insights and guidance throughout the implementation process.

**Bibliography:**

[1] M. Bostock, "Force-Directed Graph." Observable, 2020. [Online]. Available: <https://observablehq.com/@d3/force-directed-graph>. [Accessed: May 5, 2023].

[2] M. Bostock, "D3.js - Data-Driven Documents." D3.js, 2019. [Online]. Available: <https://d3js.org/>. [Accessed: May 5, 2023].

[3] Firebase, "Firebase Documentation." Firebase, 2022. [Online]. Available: <https://firebase.google.com/docs>. [Accessed: May 5, 2023].

[4] "MDN Web Docs." Mozilla Developer Network, 2023. [Online]. Available: <https://developer.mozilla.org/>. [Accessed: May 5, 2023].

**Curriculum vitae**

**Name:** Kushal Thakur

**Address:** Bhodiya, Bareli, Raisen , Mandhya Pradesh

**Phone:** +91 7999512030

**Email:** [techandgamingkushal@gmail.com](mailto:techandgamingkushal@gmail.com)

**Objective:**

To obtain a challenging and rewarding position in a dynamic organization where I can apply my technical and interpersonal skills to contribute to the success of the company.

**Education:**

* Bachelor of Computer Application, Dev Sanskriti Vishwavidyalaya, Haridwar, Uttarakhand, May 2023 (expected)
* Relevant Coursework: HTML, CSS, JS, Firebase, React, Data Structures, Algorithms, Database Systems, Software Engineering, Computer Networks
* Project Experience: Designed and Released a Media Syncing Chrome Extension, a Quiz application developed using React and API’s.

**Skills:**

* Proficient in JavaScript, C, C++, Java, Python, and SQL
* Knowledgeable in web development using HTML, CSS, JavaScript, React, D3, JQuery
* Experienced in using Agile methodologies and version control systems (Git)
* Excellent interpersonal and teamwork skills
* Strong analytical and problem-solving skills

**Certifications:**

* Gold rank on Hacker Rank in C++

**Languages:**

Hindi (native)

English (proficient)

**Interests:**

* Developing Web Applications And mobile applications using Android and iOS platforms
* Participating in hackathons and coding competitions
* Reading books and articles about emerging technologies and trends in software engineering
* Reading Self improvement Books

Declaration: I, Kushal Thakur, declare that the information provided in this curriculum vitae is true and accurate to the best of my knowledge and belief.

Signature

**2024026 : Kushal Thakur**

BCA Semester VI

Bachelor of Computer Application

Dev Sanskriti Vishwavidyalaya