**GO 1 %**

**A PROJECT REPORT**

**Submitted By**

**MOHIKA RASTOGI**

**(2100290140086)**

**AYUSH PATHAK**

**(2100290140046)**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Dr. Sangeeta Arora**

### Associate Professor



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(JUNE 2023)**

**CERTIFICATE**

Certified that **Mohika Rastogi (2100290140086), Ayush Pathak (100290140046)** have carried out the project work having “**GO 1%**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Mohika Rastogi (2100290140086)**

**Ayush Pathak (2100290140046)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Dr. Sangeeta Arora**

**Associate Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Arun Tripathi**

**Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Go 1%**

**(Mohika Rastogi, Ayush Pathak)**

**ABSTRACT**

The Go 1 % is a live project which is a website designed to enhance individual learning by incentivizing employees to learn something extra. The leaderboard website utilizes an interactive interface that allows employees to track their progress and compete with their peers. The website provides a comprehensive performance analytics dashboard that enables managers to track employee performance in real-time. The platform is designed to promote a culture of continuous learning and improvement. The leaderboard website is an innovative solution that can help organizations to enhance employee productivity and improve overall performance.

This project is a website that encourages employees to learn new things by giving them rewards and recognition. The website has a dashboard where employees can see their progress and compete with others. The website is built using different technologies, including Angular, Scala, Akka, and MongoDB, and it has several features such as Radar, Techhub, Knolx, and Helpdesk. These features provide learning resources, knowledge sharing, and technical support. The Go 1% project is a great tool for organizations that want to improve their employees' skills and productivity

**ACKNOWLEDGEMENTS**

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Sangeeta Arora** for her guidance, help, and encouragement throughout my research work. Her enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, the completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Mohika Rastogi**

**Ayush Pathak**

**LIST OF CHAPTERS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | List of Figures | | 1 |
|  | List of Tables | | 3 |
| 1 | Introduction | | 4-8 |
|  | 1.1  1.2  1.3  1.4  1.5  1.6 | Project Description  Present Challenges Facing by Organizations  Problem Statement  Solution  Overall Objective  Hardware and Software Requirement | 4  5  6  7  8  8 |
|  |
|  |
| 2 | Literature Review | | 9-14 |
|  | 2.1  2.2  2.3  2.4 | Abstract  Introduction  Literature Review  Areas of Research | 9  9  10  14 |
|  |
|  |
| 3 | Feasibility Study |  | 15-16 |
|  | 3.1  3.2  3.3  3.4 | Technical Feasibility  Operational Feasibility  Economic Feasibility  Behavioral Feasibility | 15  15  16  16 |
|  |
|  |
| 4 | Database Design |  | 17-37 |
|  | 4.1  4.2  4.3  4.4  4.5  4.6  4.7 | Flowchart  Use Case Diagram  Sequence Diagram  Activity Diagram  Class Diagram  State Diagram  Component Diagram | 17  21  24  26  31  34  36 |
|  |
|  |
|  |
|  |
| 5 | Implementation Details |  | 38-42 |
|  | 5.1  5.2  5.3  5.4  5.5  5.6  5.7  5.8 | Scala  Akka  Angular  Microservice Architecture  Database  API Testing  Development Tools  Modules in GO 1% | 38  38  39  40  40  40  40  41 |
|  |
|  |
|  |
|  |
| 6  7  8  9  10 | Design  Testing  7.1  7.2  7.3  7.4  7.5  7.6  7.7  Conclusion  Recommendation  9.1  9.2  9.3  9.4  9.5  9.6  9.7  9.8  Bibliography | Unit testing  Integration Testing  System Testing  Acceptance Testing  Software Verification and Validation  Test Procedure  Test Case  Integration of Gamification Elements  Elaboration and Social Learning Features  Personalized Learning Recommendations  Integration with External Learning Resources  Continuous Improvement and Feedback Mechanisms  Mobile Accessibility  Data Analytics and Insights  Scalability and Integration | 43-53  54-62  54  55  56  57  58  59  60  63-64  65-67  65  65  65  66  66  66  66  67  68-70 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** | **Name of Figure** | **Page No.** | |
| 4.1 | Terminal | | 18 | |
| 4.2 | Input/Output | | 18 | |
| 4.3 | Processing | | 18 | |
| 4.4 | Decision | | 19 | |
| 4.5 | Connectors | | 19 | |
| 4.6 | Flow Lines | | 19 | |
| 4.7 | Flow Chart | | 20 | |
| 4.8 | Use Case Diagram of Go 1% | | 23 | |
| 4.9 | Sequence Diagram | | 25 | |
| 4.10 | Activity | | 26 | |
| 4.11 | Forks | | 27 | |
| 4.12 | Notation of Activity Diagram | | 28 | |
| 4.13 | Activity Diagram | | 30 | |
| 4.14 | Class Diagram | | 33 | |
| 4.15 | Notation of State Diagram | | 34 | |
| 4.16 | Component | | 36 | |
| 4.17 | Node | | 36 | |
| 6.1 | Dashboard | | 44 | |
| 6.2 | Monthly Contributions on Dashboard | | 45 | |
| 6.3 | Profile | | 46 | |
| 6.4 | Redeemed Rewards | | 47 | |
| 6.5 | Radar | | 48 | |
| 6.6 | Track Learning Status on Radar | | 49 | |
| 6.7 | Leaderboard | | 50 | |
| 6.8 | Wall of Fame on Leaderboard | | 50 | |
| 6.9 | Rewards on Leaderboard | | 51 | |
| 6.10 | Knolx | | 52 | |
| 6.11 | Login/Registration | | 53 | |

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** | **Name of Figure** | **Page No.** | |
| 1.1 | Hardware Specifications | | 8 | |
| 1.2 | Software Specifications | | 8 | |

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT DESCRIPTION**

The Go 1% project is an innovative website designed to enhance individual learning, foster employee engagement, and drive continuous improvement within organizations. By incentivizing employees to go beyond their limits and promoting a culture of lifelong learning, the platform aims to improve employee productivity and overall performance.

At the core of the Go 1% website is a dynamic leaderboard that serves as a central hub for tracking progress and promoting healthy competition among employees. This interactive interface allows users to monitor their individual achievements, compare their progress with their peers, and earn recognition for their efforts. The leaderboard showcases rankings, badges, and rewards, creating a sense of achievement and motivation for employees to continually strive for growth.

The platform offers a range of modules and features that support employee learning and development. The Dashboard module provides a comprehensive overview of the user's profile, displaying key information such as points earned, learning activities completed, and available rewards. This enables employees to visualize their progress and set goals for their personal and professional growth. The Dashboard also includes a graphical representation of user contributions, highlighting the positive impact of their efforts within the organization.

To bring the Go 1% website to life, it is built using a combination of technologies. The front-end development is implemented using Angular, a popular framework known for its dynamic and responsive user interface. The back-end development is powered by Scala, a robust programming language, with the Akka toolkit providing a scalable and fault-tolerant framework for building reactive applications. The platform follows a microservices architecture, allowing for modular and independent services that can be deployed and scaled individually. MongoDB serves as the primary database storage technology, providing scalability and flexibility for data management.

In conclusion, the Go 1% project is a powerful tool for organizations seeking to enhance employee productivity and improve overall performance. By creating a culture of continuous learning, promoting healthy competition, and providing comprehensive performance analytics, the platform empowers employees to develop new skills, stay motivated, and contribute to the organization's success. With its diverse modules and features, the Go 1% website offers a comprehensive solution to support individual learning, foster collaboration, and drive positive change within organizations.

**1.2 PRESENT CHALLENGES FACED BY ORGANIZATIONS**

1. **Lack of Employee Engagement**: Many organizations struggle with low employee engagement, where employees may feel disinterested or disconnected from their work. The Go 1% tool promotes engagement by incentivizing learning and creating a competitive environment. The leaderboard fosters healthy competition among employees, motivating them to actively participate and improve their skills.
2. **Limited Learning Opportunities:** Organizations often face challenges in providing diverse and accessible learning opportunities for their employees. The Go 1% platform offers a wide range of learning resources through modules such as Techhub, Knolx, and Radar. Employees can access tutorials, courses, and documentation to acquire new knowledge and skills, enabling continuous learning and development.
3. **Lack of Performance Visibility:** Tracking employee performance in real-time can be a challenge for organizations. The Go 1% leaderboard provides comprehensive performance analytics, allowing managers to monitor employee progress and contributions. This visibility helps identify top performers, recognize achievements, and provide targeted support to employees who may need assistance.
4. **Limited Collaboration and Knowledge Sharing:** Organizations often struggle with facilitating collaboration and knowledge sharing among employees. The Go 1% platform addresses this challenge through features like Knolx and Helpdesk. Employees can share their expertise, conduct presentations and webinars, and participate in forums, fostering a collaborative learning environment.
5. **Difficulty in Promoting a Learning Culture:** Establishing a culture of continuous learning can be a significant challenge for organizations. The Go 1% tool promotes a learning culture by incentivizing and recognizing employee learning efforts. The leaderboard and rewards system create a sense of achievement and motivate employees to engage in continuous learning, resulting in a positive and growth-oriented work environment.
6. **Tracking and Assessing Progress:** Monitoring employee progress and assessing their learning outcomes can be time-consuming and complex. The Go 1% platform simplifies this process by providing a centralized dashboard that tracks individual progress, points earned, and contributions. This data can be used for performance evaluations, identifying areas for improvement, and recognizing employee accomplishments.
7. **Employee Retention and Talent Development:** Organizations often face challenges in retaining talented employees and nurturing their skills. The Go 1% platform offers a gamified approach to learning and development, which can increase employee satisfaction and engagement. By providing opportunities for growth and recognition, the tool contributes to talent development and helps retain valuable employees.

Overall, the Go 1% management or leaderboard tool addresses common challenges faced by organizations, such as employee engagement, learning opportunities, performance visibility, collaboration, promoting a learning culture, progress tracking, and talent development. By leveraging these features, organizations can enhance employee productivity, improve overall performance, and create a positive and conducive work environment.

**1.3 PROBLEM STATEMENT**

Many organizations struggle to create a culture of continuous learning and improvement among their employees. Existing approaches fail to incentivize employees to engage in ongoing learning activities and limit opportunities for collaboration and knowledge sharing. Additionally, organizations face challenges in tracking employee performance and providing comprehensive visibility into individual progress. As a result, employee engagement and productivity suffer, hindering the organization's ability to adapt to changing technologies and market demands. There is a need for a solution that promotes active learning, fosters collaboration, and offers real-time performance tracking to enhance employee engagement, productivity, and overall organizational performance.

**1.4 SOLUTION**

The proposed solution to address the challenges mentioned above is the implementation of the Go 1% management or leaderboard tool. This tool is designed to promote a culture of continuous learning and improvement within organizations by incentivizing employees to engage in ongoing learning activities.

The Go 1% platform offers a range of features and modules that facilitate learning, collaboration, and performance tracking. Employees can access learning resources through modules such as Techhub, Knolx, and Radar, which provide tutorials, courses, and a categorized technology landscape for effective learning. The leaderboard module fosters healthy competition by allowing employees to track their progress, compare their achievements with peers, and earn recognition for their learning efforts.

The platform also provides managers with a comprehensive performance analytics dashboard, enabling real-time monitoring of employee progress. This visibility allows for targeted support and recognition, further motivating employees to actively participate in the learning process. The Helpdesk module offers technical support, ensuring that employees can overcome any learning challenges they may face.

By implementing the Go 1% management or leaderboard tool, organizations can create a positive and engaging work environment that encourages continuous learning, collaboration, and skill development. This solution promotes employee engagement, improves productivity, and enables organizations to adapt to changing technologies and market demands more effectively. Ultimately, it contributes to the overall growth and success of the organization by harnessing the full potential of its workforce.

**1.5 OVERALL OBJECTIVE**

The overall objective of the Go 1% management or leaderboard project is to create a culture of continuous learning and improvement within organizations. By implementing this tool, the project aims to incentivize employees to engage in ongoing learning activities, foster collaboration, and provide real-time performance tracking. The objective is to enhance employee engagement, productivity, and overall organizational performance by promoting a positive work environment that values and supports continuous learning and skill development.

**1.6 HARDWARE & SOFTWARE SPECIFICATIONS**

**Table: 1.1 HARDWARE SPECIFICATIONS**

|  |  |
| --- | --- |
|  |  |
| Processor | 11th Gen Intel(R) Core (TM) i3 |
|  |  |
| System Type | 32-bit operating system |
|  |  |
| Speed | 3.00 GHz |
|  |  |
| Hard Disk Space | 256 GB |
|  |  |
| RAM Memory | 4 GB |
|  |  |
| Operating System | Windows 8 |
|  |  |

**Table: 1.2 SOFTWARE SPECIFICATIONS**

|  |  |  |
| --- | --- | --- |
|  | |  |
| Database Management System | | POSTMAN, MongoDB |
|  | |  |
| Development Environment (IDE) | | IntelliJ Community IDE |
|  | |  |
| Code Editor | | IntelliJ |
|  |  |  |

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 ABSTRACT**

The Go 1 % is a live project which is a website designed to enhance individual learning by incentivizing employees to learn something extra. The leaderboard website utilizes an interactive interface that allows employees to track their progress and compete with their peers. The website provides a comprehensive performance analytics dashboard that enables managers to track employee performance in real-time. The platform is designed to promote a culture of continuous learning and improvement. The leaderboard website is an innovative solution that can help organizations to enhance employee productivity and improve overall performance.

This project is a website that encourages employees to learn new things by giving them rewards and recognition. The website has a dashboard where employees can see their progress and compete with others. The website is built using different technologies, including Angular, Scala, Akka, and MongoDB, and it has several features such as Radar, Techhub, Knolx, and Helpdesk. These features provide learning resources, knowledge sharing, and technical support. The Go 1% project is a great tool for organizations that want to improve their employees' skills and productivity.

**2.2 INTRODUCTION**

In today's rapidly changing business landscape, organizations recognize the importance of continuous learning and development to remain competitive. The Go 1% project aims to promote a culture of continuous learning by incentivizing employees through a website and leaderboard system. This literature review explores relevant research on incentivizing learning, the role of technology in fostering learning cultures, the impact of competition and recognition on employee performance, and potential areas of research for the Go 1% project.

**2.3 LITERATURE REVIEW**

Literature review is an expressive study based on the detailed review of earlier pertinent studies related to the various concepts of online shopping to discover the concept of online shopping. It highlights the status of online shopping, importance and problems of online shopping, factors affecting online shopping and a critical review of the privacy and security issues in online shopping.

**2.3.1 STATUS OF LEADERBOARD PLATFORMS IN PRESENT BUSINESSES**

Leaderboard platforms have gained significant popularity in modern businesses as a means to enhance employee engagement, foster competition, and drive performance. This literature review examines the current status of leaderboard platforms in present businesses, exploring their benefits, challenges, and the impact they have on employee motivation and performance. By analysing existing research and industry practices, this review aims to provide insights into the effectiveness and trends of leaderboard platforms in contemporary business environments.

1. **Benefits of Leaderboard Platforms:**

Numerous studies highlight the benefits of leaderboard platforms in businesses. Firstly, leaderboard platforms create a sense of healthy competition among employees, driving them to excel and achieve higher performance levels. Research indicates that competition can increase motivation and productivity, leading to improved individual and team outcomes. Additionally, leaderboard platforms promote transparency and visibility, allowing employees to track their progress, compare their performance with peers, and gain a sense of achievement and recognition. Furthermore, these platforms facilitate knowledge sharing, collaboration, and the development of a learning culture within organizations.

1. **Challenges and Limitations:**

While leaderboard platforms offer various benefits, they also come with challenges and limitations. One common challenge is ensuring the fairness and accuracy of performance metrics used in the rankings. Biases or subjective evaluations can undermine the credibility and effectiveness of the platform. Moreover, leaderboard platforms may inadvertently create a negative work environment if not implemented properly. Excessive competition or a lack of recognition for efforts can lead to demotivation and stress among employees. It is crucial to address these challenges through careful design, regular evaluation, and timely feedback mechanisms.

1. **Adoption and Implementation in Business Environments:**

The adoption of leaderboard platforms varies across different industries and organizations. Research indicates that industries with high-performance cultures, such as sales, technology, and gaming, are more likely to embrace leaderboard platforms as they align well with their competitive nature. Moreover, organizations that prioritize continuous learning and employee development tend to integrate leaderboard platforms to encourage skill acquisition and knowledge sharing. However, the successful implementation of leaderboard platforms requires a supportive organizational culture, effective communication, and alignment with overall business goals.

1. **Gamification and Leaderboard Platforms:**

Gamification, the application of game elements in non-game contexts, is often intertwined with leaderboard platforms. Research suggests that gamification features, such as badges, rewards, and levels, can enhance the engagement and motivation of employees on leaderboard platforms. By incorporating game-like elements, organizations can create an immersive and enjoyable experience for employees, leading to increased participation and sustained usage of the platform. However, careful consideration should be given to strike a balance between intrinsic and extrinsic motivators to maintain long-term engagement.

1. **Future Trends and Directions:**

The future of leaderboard platforms in businesses is expected to witness advancements in customization and personalization. Tailoring the platform to individual preferences, goals, and learning styles can enhance its effectiveness and relevance. Furthermore, the integration of artificial intelligence and machine learning algorithms may enable predictive analytics, providing personalized recommendations for learning resources and development opportunities. Additionally, the integration of social and collaborative features in leaderboard platforms can foster a sense of community and collective learning.

**2.3.2 BENEFITS OF HAVING LEADERBOARD SYSTEMS IN ORGANIZATIONS**

Leaderboard platforms offer several advantages for organizations aiming to enhance employee learning and performance:

* **Encouraging Competition**: Leader board platforms promote healthy competition among employees, fostering motivation and driving performance improvements. By providing a transparent and visible ranking system, employees are incentivized to excel and outperform their peers.
* **Recognition and Reward**: Leader boards enable organizations to recognize and reward top performers publicly. By showcasing individual achievements and contributions, leaderboard platforms reinforce a culture of recognition, boosting employee morale and job satisfaction.
* **Transparency and Visibility**: Leader boards provide real-time visibility into employee performance, allowing managers to track progress and identify areas for improvement. Transparent performance data promotes accountability and facilitates targeted coaching and support.
* **Collaboration and Knowledge Sharing:** Leaderboard platforms often incorporate features such as forums, knowledge sharing portals, and collaborative spaces. These tools facilitate cross-team collaboration, enabling employees to share insights, best practices, and valuable resources.

**2.3.3 PROBLEMS OF LEADERBOARD PLATFORMS**

Leaderboard platforms, despite their benefits, can also face certain problems and challenges. Here are some common issues associated with leaderboard platforms:

1. **Unintended Consequences**: Leaderboard platforms can inadvertently lead to negative consequences if not carefully designed and implemented. Excessive competition and a focus solely on rankings can create a hostile work environment, fostering cutthroat behaviours and undermining collaboration. It is essential to strike a balance between healthy competition and fostering a supportive team environment.
2. **Biases and Inequities**: Leader boards rely on specific metrics and performance indicators to rank individuals or teams. However, if these metrics are biased or do not accurately reflect the desired outcomes, it can lead to unfair rankings. Biases can arise from subjective evaluations, inconsistent criteria, or inadequate data collection methods. It is crucial to ensure that the metrics used are fair, objective, and aligned with the organization's goals.
3. **Demotivation and Disengagement**: While competition can be motivating for some individuals, it may demotivate and disengage others. Employees who consistently rank lower on leader boards may feel discouraged or undervalued, leading to decreased motivation and job satisfaction. Moreover, leaderboard platforms that solely focus on rankings may overshadow other important aspects of performance and contributions.
4. **Lack of Context and Complexity**: Leader boards often simplify complex skills and performance criteria into single metrics or rankings. This oversimplification may fail
5. to capture the full range of abilities, expertise, and contributions of individuals or teams. Certain valuable qualities, such as creativity, collaboration, or problem-solving skills, may not be adequately represented in a leaderboard format.
6. **Gaming the System**: In competitive environments, some individuals may attempt to manipulate the leaderboard rankings to their advantage. This can involve unethical practices, such as data manipulation, sabotaging others' performance, or exploiting loopholes in the evaluation process. Such behaviours undermine the integrity of the leaderboard platform and can create a culture of distrust and unfairness.
7. **Short-term Focus and Learning Narrowness**: Leaderboard platforms may inadvertently encourage employees to prioritize short-term goals and outcomes to secure higher rankings. This narrow focus may hinder long-term learning, personal development, and exploration of innovative ideas. It is crucial to balance immediate performance with continuous learning and development to avoid sacrificing broader organizational goals.
8. **Lack of Recognition for Non-Quantifiable Contributions**: Leader boards primarily focus on quantifiable metrics, which may overlook or undervalue non-quantifiable contributions. Soft skills, collaboration, mentoring, or efforts that support team dynamics are often difficult to quantify and may go unnoticed in a leaderboard system. It is important to incorporate mechanisms that recognize and appreciate diverse contributions beyond measurable outcomes.

Addressing these problems requires careful consideration and design. Organizations should focus on creating a supportive and inclusive work environment that values collaboration, recognizes various forms of contributions, and places the leaderboard platform in the broader context of overall performance management. Regular evaluation, feedback mechanisms, and flexibility in metrics can help mitigate the challenges associated with leaderboard platforms and promote their effective and fair use in organizations.

**2.4 AREAS OF RESEARCH**

To enhance the effectiveness of leaderboard platforms, several areas of research merit exploration:

1. **Designing Inclusive Metrics**: Research should focus on developing comprehensive metrics that capture both quantitative and qualitative aspects of employee performance. By incorporating diverse indicators, organizations can promote a more holistic evaluation process.
2. **Mitigating Biases**: Studies can explore strategies to minimize biases in leaderboard rankings. This includes evaluating the impact of different evaluation criteria, implementing bias detection algorithms, and providing training to reduce unconscious biases in the performance assessment process.
3. **Personalization and Individual Goals**: Research should investigate ways to personalize leaderboard platforms to align with individual learning goals and aspirations. Customizable dashboards and adaptive algorithms can help tailor the leaderboard experience to each employee's unique needs.
4. **Balancing Competition and Collaboratio**n: Studies can explore methods to strike a balance between competition and collaboration within leaderboard platforms. This involves designing features that foster teamwork, encourage knowledge sharing, and emphasize collective achievements alongside individual ranking.

**CHAPTER 3**

**FEASIBILITY STUDY**

A feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn’t profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

**3.1 TECHNICAL FEASIBILITY**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team can convert the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. It includes finding out technologies for the project, both hardware and software. For a virtual assistant, the user must have a microphone to convey their message and a speaker to listen when the system speaks. These are very cheap nowadays and everyone generally possesses them. Besides, the system needs an internet connection. While using the assistant, make sure you have a steady internet connection. It is also not an issue in this era where almost every home or office has Wi-Fi.

**3.2 OPERATIONAL FEASIBILITY**

This assessment involves undertaking a study to analyse and determine whether and how well— the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development. It is the ease and simplicity of operation of the proposed system. The system does not require any special skill set for users to operate it. This shows the management and organizational structure of the project. This project is not built by a team. The management tasks are all to be carried out by a single person. That won’t create any management issues and will increase the feasibility of the project.

**3.3 ECONOMICAL FEASIBILITY**

In the Economic Feasibility study cost and benefit of the project are analysed. This means under this feasibility study a detailed analysis is carried out will be the cost of the project for development which includes all required costs final development hardware and software resources required, design and development costs and operational costs, and so on.

After that, it is analysed whether the project will be beneficial in terms of finance for the organization or not. we find the total cost and benefit of the proposed system over the current system. For this project, the main cost is the documentation cost. Users also would have to pay for the microphone and speakers. Again, they are cheap and available. As far as maintenance is concerned, Assistant won’t cost too much.

**3.4 BEHAVIOURAL FEASIBILITY**

It evaluates and estimates the user attitude or behaviour toward the development of the new system. It helps in determining if the system requires special effort to educate, retrain, transfer, and change an employee's job status on new ways of conducting business. Establishing the cost-effectiveness of the proposed system i.e., if the benefits do not outweigh the costs, then it is not worth going ahead. In the fast-paced world today there is a great need for online social networking facilities.

Thus, the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

**CHAPTER 4**

**DATABASE DESIGN**

A properly designed database provides you with access to up-to-date, accurate information. Because a correct design is essential to achieving your goals in working with a database, investing the time required to learn the principles of good design makes sense. In the end, you are much more likely to end up with a database that meets your needs and can easily accommodate change. This article provides guidelines for planning a desktop database. You will learn how to decide what information you need, how to divide that information into the appropriate tables and columns, and how those tables relate to each other. You should read this article before you create your first desktop database.

Database design can be generally defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of an enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are greatly influenced in terms of disk storage space. Therefore, there has to be a brilliant concept for designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

The main objectives behind database designing are to produce physical and logical design models of the proposed database system. To elaborate on this, the logical model is primarily concentrated on the requirements of data and the considerations must be made in terms of monolithic considerations hence the stored physical data must be stored independent of the physical conditions. On the other hand, the physical database design model includes a translation of the logical design model of the database by keeping control of physical media using hardware resources and software systems such as Database Management System (DBMS).

**4.1 FLOW CHART**

A flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols that are connected among them to indicate the flow of information and processing.

The process of drawing a flowchart for an algorithm is known as “flowcharting”. **Basic Symbols used in Flowchart Designs**

**Terminal:** The oval symbol indicates Start, Stop, and Halt in a program’s logic flow. A pause/halt is generally used in a program logic under some error conditions. The terminal is the first and last symbol in the flowchart.



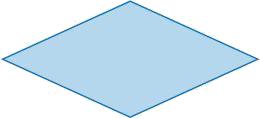
**Fig: 4.1 Terminal**

**Input/Output:** A parallelogram denotes any function of input/output type. Program instructions that take input from input devices and display output on output devices are indicated with parallelograms in a flowchart.

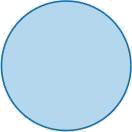
**Fig: 4.2 Input/Output**

**Processing:** A box represents arithmetic instructions. All arithmetic processes such as adding, subtracting, multiplication, and division are indicated by action or process symbol.

**Fig: 4.3 Processing**

**Decision** Diamond symbol represents a decision point. Decision-based operations such as yes/no questions or true/false are indicated by diamonds in the flowchart.

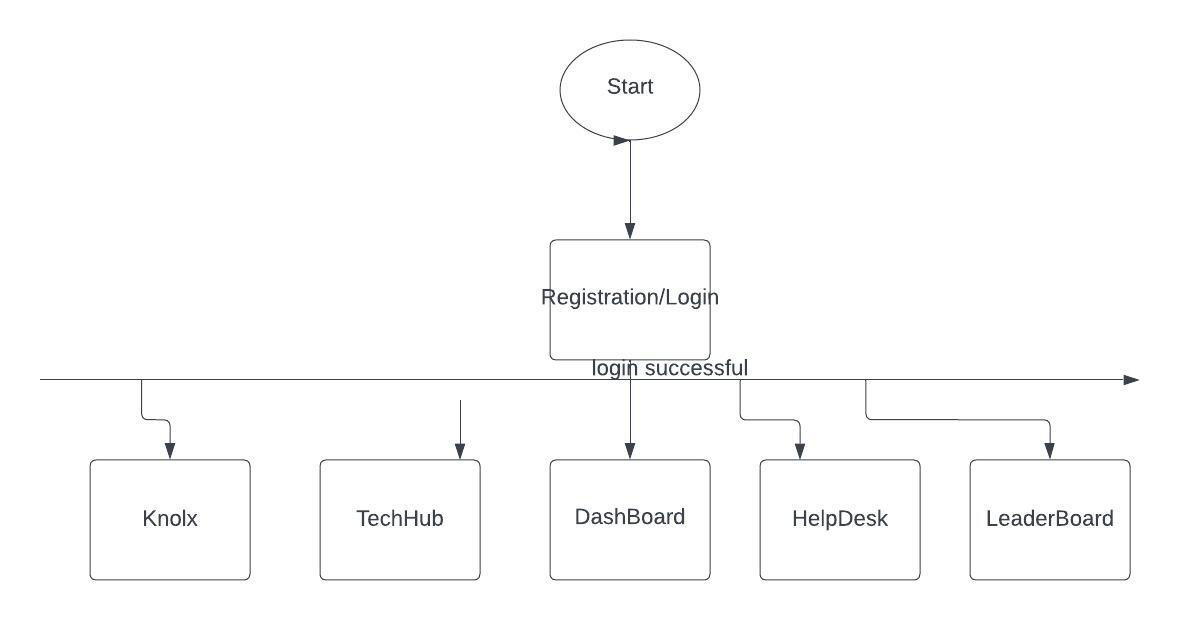
**Fig: 4.4 Decision**

**Connectors:** Whenever the flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.

**Fig: 4.5 Connectors**

**Flow lines:** Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of the flow of control and the relationship among different symbols of the flowchart.

**Fig: 4.6 Flow lines**



**Fig: 4.7 Flow Chart**

**4.2 USE CASE DIAGRAM**

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.

The purpose of a use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as the other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from the other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

**In brief, the purposes of use case diagrams can be said to be as follows −**

Used to gather the requirements of a system.

Used to get an outside view of a system.

Identify the external and internal factors influencing the system.

Show the interaction among the requirements actors.

**Use case diagram components.**

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

**Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

**System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

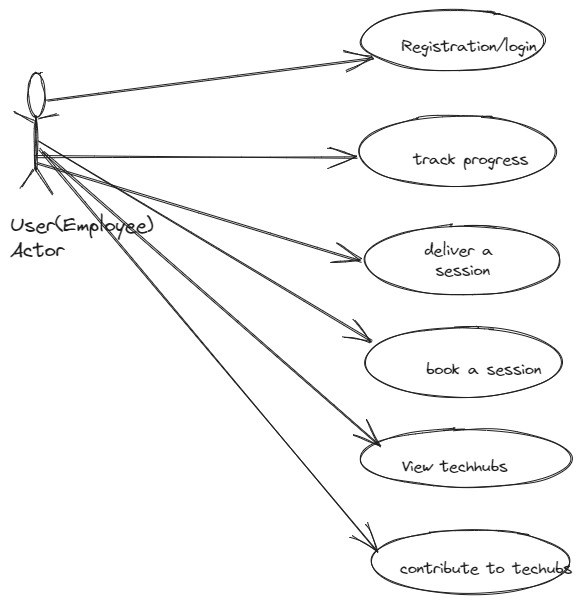
**Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

**Use case diagram symbols and notation.**

The notation for a use case diagram is pretty straight forward and doesn't involve as many types of symbols as other UML diagrams.

**Use cases:** Horizontally shaped ovals that represent the different uses that a user might have.

* **Actors:** Stick figures that represent the people actually employing the use cases.
* **Associations:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
* **System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.
* **Packages:** A UML shape that allows you to put different elements into groups. Just as with component diagrams, these groupings are represented as file folders.

****

**Fig 4.8: Use Case Diagram of Go 1%**

**4.3 SEQUENCE DIAGRAM**

A **sequence diagram** or **s**ystem sequence diagram (SSD) shows object interactions arranged in time sequence in the field of [software engineering.](https://en.wikipedia.org/wiki/Software_engineering) It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of scenario. Sequence diagrams are typically associated with use case realizations in the [logical](https://en.wikipedia.org/wiki/4+1_architectural_view_model) view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

For a particular scenario of a [use case,](https://en.wikipedia.org/wiki/Use_case) the diagrams show the events that external actors generate, their order, and possible inter-system events. All [systems](https://en.wikipedia.org/wiki/Software_system) are treated as a [black box;](https://en.wikipedia.org/wiki/Black_box) the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be done for the main success scenario of the [use case,](https://en.wikipedia.org/wiki/Use_case) and frequent or complex alternative scenarios.



To model high-level interaction among active objects within a system.

To model interaction among objects inside a collaboration realizing a use case.

It either models’ generic interactions or some certain instances of interaction.

**Sequence Diagram Notations –**

**i. Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram

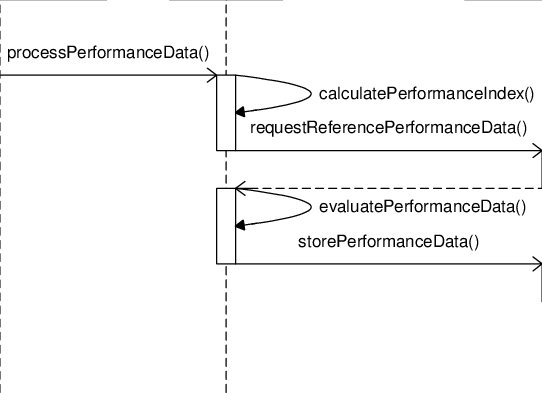
1. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
2. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
3. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

**Uses of sequence diagrams –**

i. Used to model and visualise the logic behind a sophisticated function, operation or procedure.

1. They are also used to show details of UML use case diagrams.
2. Used to understand the detailed functionality of current or future systems.
3. Visualise how messages and tasks move between objects or components in a system.

**For Performance Evaluation**

****

**Fig 4.9: Sequence Diagra**

**4.4 ACTIVITY DIAGRAM**

The activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioural diagram.

**Components of an Activity Diagram**

Following are the component of an activity diagram:

**Activities**

The categorization of behaviour into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes.

The control flow of activity is represented by control nodes and object nodes that illustrates the objects used within an activity. The activities are initiated at the initial node and are terminated at the final node.



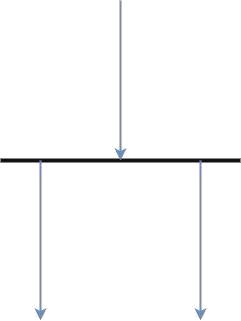
**Fig: 4.10 Activity**

**Activity partition /swim lane**

The swim lane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swim lane in the activity diagram. But it is used to add more transparency to the activity diagram.

**Forks**

Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever a data is received at an inward edge, it gets copied and split crossways various outward edges. It split a single inward flow into multiple parallel flows



**Fig: 4.11 Forks**

**Join Nodes**

Join nodes are the opposite of fork nodes. A Logical AND operation is performed on all of the inward edges as it synchronizes the flow of input across one single output (outward) edge.

**Pins**

It is a small rectangle, which is attached to the action rectangle. It clears out all the messy and complicated thing to manage the execution flow of activities. It is an object node that precisely represents one input to or output from the action.

**Notation of an Activity diagram**

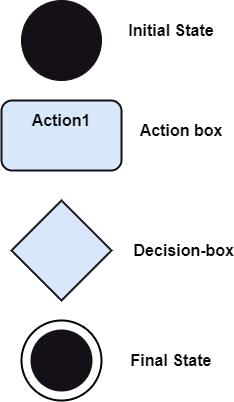
Activity diagram constitutes following notations:

**Initial State:** It depicts the initial stage or beginning of the set of actions.

**Final State:** It is the stage where all the control flows and object flows end.

**Decision Box:** It makes sure that the control flow or object flow will follow only one path.

**Action Box:** It represents the set of actions that are to be performed.



**Fig: 4.12 Notation of Activity Diagr**

**Why use Activity Diagram?**

An event is created as an activity diagram encompassing a group of nodes associated with edges. To model the behaviour of activities, they can be attached to any modelling element. It can model use cases, classes, interfaces, components, and collaborations.

It mainly models processes and workflows. It envisions the dynamic behaviour of the system as well as constructs a runnable system that incorporates forward and reverse engineering. It does not include the message part, which means message flow is not represented in an activity diagram.

It is the same as that of a flowchart but not exactly a flowchart itself. It is used to depict the flow between several activities.

**How to draw an Activity Diagram?**

An activity diagram is a flowchart of activities, as it represents the workflow among various activities. They are identical to the flowcharts, but they themselves are not exactly the flowchart. In other words, it can be said that an activity diagram is an enhancement of the flowchart, which encompasses several unique skills.

Since it incorporates swim lanes, branching, parallel flows, join nodes, control nodes, and forks, it supports exception handling. A system must be explored as a whole before drawing an activity diagram to provide a clearer view of the user. All of the activities are explored after they are properly analysed for finding out the constraints applied to the activities. Each and every activity, condition, and association must be recognized.

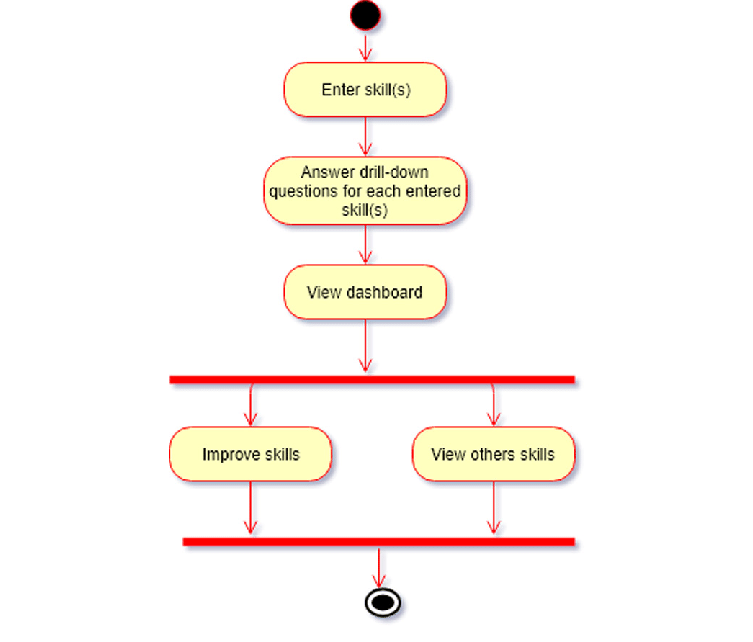
After gathering all the essential information, an abstract or a prototype is built, which is then transformed into the actual diagram.

**Following are the rules that are to be followed for drawing an activity diagram:**

A meaningful name should be given to each and every activity.

Identify all of the constraints.

Acknowledge the activity associations



**Fig: 4.13 Activity Diagr**

**4.5 CLASS DIAGRAM**

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

**Purpose of Class Diagrams**

The main purpose of class diagrams is to build a static view of an application. It is the only diagram that is widely used for construction, and it can be mapped with object-oriented languages. It is one of the most popular UML diagrams. Following are the purpose of class diagrams given below:

It analyses and designs a static view of an application.

It describes the major responsibilities of a system.

It is a base for component and deployment diagrams. It incorporates forward and reverse engineering

**Benefits of Class Diagrams**

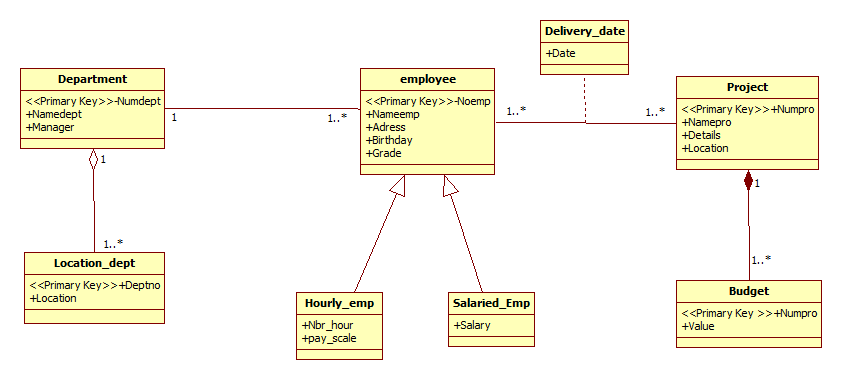
1. It can represent the object model for complex systems.
2. It reduces the maintenance time by providing an overview of how an application is structured before coding.
3. It provides a general schematic of an application for better understanding.
4. It represents a detailed chart by highlighting the desired code, which is to be programmed.
5. It is helpful for the stakeholders and the developers.

**How to draw a Class Diagram?**

The class diagram is used most widely to construct software applications. It not only represents a static view of the system but also all the major aspects of an application. A collection of class diagrams as a whole represents a system.

Some key points that are needed to keep in mind while drawing a class diagram are given below:

1. To describe a complete aspect of the system, it is suggested to give a meaningful name to the class diagram.
2. The objects and their relationships should be acknowledged in advance.
3. The attributes and methods (responsibilities) of each class must be known.
4. A minimum number of desired properties should be specified as a greater number of the unwanted property will lead to a complex diagram.
5. Notes can be used as and when required by the developer to describe the aspects of a diagram.
6. The diagrams should be redrawn and reworked as many times to make it correct before producing its final version.



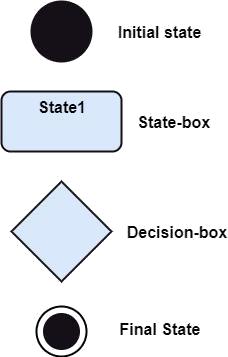
**4.6 STATE DIAGRAM**

The state machine diagram is also called the State chart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behaviour. It models the behaviour of a class, a subsystem, a package, and a complete system.

It tends out to be an efficient way of modelling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

**Notation of a State Machine Diagram**

Following are the notations of a state machine diagram enlisted below:



**Fig: 4.15 Notation of State Diagram**

**a. Initial state:** It defines the initial state (beginning) of a system, and it is represented by a black filled circle.

**b. Final state:** It represents the final state (end) of a system. It is denoted by a filled circle present within a circle.

**c. Decision box:** It is of diamond shape that represents the decisions to be made on the basis of an evaluated guard.

**d. Transition:** A change of control from one state to another due to the occurrence of some event is termed as a transition. It is represented by an arrow labelled with an event due to which the change has ensued.

**e. State box:** It depicts the conditions or circumstances of a particular object of a class at a specific point of time. A rectangle with round corners is used to represent the state box.

**How to Draw a State Machine Diagram?**

The state machine diagram is used to portray various states underwent by an object. The change in one state to another is due to the occurrence of some event. All of the possible states of a particular component must be identified before drawing a state machine diagram.

The primary focus of the state machine diagram is to depict the states of a system. These states are essential while drawing a state transition diagram. The objects, states, and events due to which the state transition occurs must be acknowledged before the implementation of a state machine diagram.

**Following are the steps that are to be incorporated while drawing a state machine diagram:**

1. A unique and understandable name should be assigned to the state transition that describes the behaviour of the system.
2. Out of multiple objects, only the essential objects are implemented.
3. A proper name should be given to the events and the transitions.

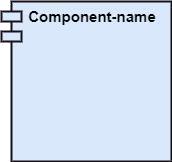
**4.7 COMPONENT DIAGRAM**

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviour is explained by the provided and required interfaces.

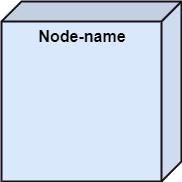
**Notations Of a Component Diagram**

a) A component



**Fig: 4.16 Component**

b) A node



b) A node

A blue cube with black text

Description automatically generated with low confidence

**Fig: 4.17 Node**

**How to Draw a Component Diagram?**

The component diagram is helpful in representing the physical aspects of a system, which are files, executables, libraries, etc. The main purpose of a component diagram is different from that of other diagrams. It is utilized in the implementation phase of any application.

Once the system is designed employing different UML diagrams, and the artifacts are prepared, the component diagram is used to get an idea of implementation. It plays an essential role in implementing applications efficiently.

**Following are some artifacts that are needed to be identified before drawing a component diagram:**

1. What files are used inside the system?
2. What is the application of relevant libraries and artifacts?
3. What is the relationship between the artifacts?

**Following are some points that are needed to be kept in mind after the artifacts are identified:**

1. Using a meaningful name to ascertain the component for which the diagram is about to be drawn.
2. Before producing the required tools, a mental layout is to be made.
3. To clarify the important points, notes can be incorporated.

**CHAPTER 5**

**IMPLEMENTATION DETAILS**

The Go 1% project utilizes several technologies and frameworks to implement its features and functionalities. Here are some implementation details that can be included in the project report:

**5.1 Scala**

Scala is a statically typed programming language that combines object-oriented programming (OOP) and functional programming (FP) paradigms. It is designed to be expressive, concise, and scalable, making it a popular choice for building robust and high-performance applications. Scala is used for back-end development.

**5.2 Akka**

Akka is a powerful toolkit and runtime environment for building highly concurrent, distributed, and fault-tolerant applications. It is implemented in Scala but also offers support for Java and other JVM languages. Akka is built on the principles of the Actor Model, which provides a lightweight and efficient approach to concurrent programming.

Key features and concepts of Akka include:

* **Actors**: Akka's core abstraction is the "actor." Actors are concurrent entities that encapsulate state and behaviour. They communicate with each other by exchanging messages, enabling asynchronous and non-blocking interactions. Actors provide a high level of concurrency and isolation, making it easier to reason about complex systems.
* **Concurrency and Scalability:** Akka is designed to handle massive concurrency and scale across multiple nodes in a distributed environment. It provides tools and abstractions to manage concurrent computations, such as supervision hierarchies, routing strategies, and load balancing.
* **Fault Tolerance**: Akka promotes the development of fault-tolerant systems by providing built-in mechanisms for handling failures and recovering from errors. Actors can be supervised, and failures are isolated within individual actors, preventing cascading failures across the system.
* **Remoting and Clustering**: Akka supports transparent remote communication between actors, allowing them to interact seamlessly across network boundaries. It also provides clustering capabilities, enabling the formation of fault-tolerant and elastic clusters of actors.

In the context of the Go 1% platform, Scala and Akka are used together to build the back-end infrastructure. Scala's expressive syntax and powerful abstractions make it well-suited for implementing the business logic, while Akka provides the concurrency model and fault-tolerance mechanisms necessary for building a scalable and responsive system. This combination allows for the development of a robust and high-performing platform that can handle the concurrent interactions and distributed nature of the application.

**5.3 ANGULAR**

This topic can help you understand Angular: what Angular is, what advantages it provides, and what you might expect as you start to build your applications.

Angular is a development platform, built on [TypeScript](https://www.typescriptlang.org/). As a platform, Angular includes:

A component-based framework for building scalable web applications

A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more

A suite of developer tools to help you develop, build, test, and update your code

with Angular, you're taking advantage of a platform that can scale from single-developer projects to enterprise-level applications. Angular is designed to make updating as straightforward as possible, so take advantage of the latest developments with minimal effort. Best of all, the Angular ecosystem consists of a diverse group of over 1.7 million developers, library authors, and content creators.

**5.4 Microservices Architecture**

The project follows a microservices architecture, which allows for the development of modular and independent services.

Each module, such as Dashboard, Radar, Leaderboard, Techhub, Knolx, Helpdesk, is implemented as a separate microservice.

Microservices communicate with each other through APIs and can be deployed and scaled independently.

**5.5 Database**

MongoDB is used as the primary database storage technology.

It provides a scalable and flexible document-based data store, suitable for storing various types of data.

**5.6 API Testing**

Postman tool is used for API testing, documentation, and collaboration.

It allows developers to test and verify the functionality of the APIs.

**5.7 Development Tools**

IntelliJ IDEA is an integrated development environment (IDE) commonly used for Scala development.

Eclipse IDE can also be used for Scala development, providing features for code editing and debugging.

Git is a version control system used for managing source code and collaboration.

These are some of the implementation details related to the technologies and tools used in the project. You can further expand on each technology, explaining its benefits and how it contributes to the overall functionality of the Go 1% platform. Additionally, you can include any specific libraries or frameworks used within each module, such as libraries for data visualization, authentication, or third-party integrations.

**5.8 Modules used in GO 1%**

**5.8.1 Dashboard**

The Dashboard module is the main interface of the Go 1% website, providing employees with an interactive dashboard to track their progress and compete with their peers. Within the module, users can view their profile information and see how many points they have collected through their learning and development activities. The dashboard also displays a description of the rewards available and the number of points required to unlock them.

In addition, the dashboard includes a graphical representation of the user's progress, highlighting their contributions to any activities in the organization that have brought about positive change and learning. This feature enables users to see the impact of their efforts and motivates them to continue their learning journey. Overall, the Dashboard module serves as a central hub for users to monitor their progress, track their accomplishments, and stay engaged with their personal and professional development goals. This page also has a search bar where we can search any other employee from our organization.

**5.8.2 Radar**

The Radar module of Go 1% website categorizes various technologies, frameworks, tools, and techniques into Adopt, Trial, Assess, and Hold categories. It helps users to understand which technologies are widely adopted and recommended as best practices, which ones are being tested, which ones require assessment, and which ones are not recommended for use. This module provides a visual representation of the technology landscape, making it easier for users to identify the most relevant and effective technologies to use for their personal and professional development.

**5.8.3 Leaderboard**

The leaderboard module allows employees to compete with their peers and earn recognition for their achievements. It includes features such as rankings, badges, and rewards.

**5.4.4 Techhub**

The techhub module is a repository of learning resources, including tutorials, courses, and documentation. It helps employees to discover and access relevant learning materials.

**5.4.5 Knolx**

The knolx module is a knowledge sharing platform that enables employees to share their expertise with their peers. It includes features such as presentations, webinars, and forums.

**5.8.6 Helpdesk**

The helpdesk module provides employees with support and assistance for any technical issues or learning challenges they may face.

**CHAPTER 6**

**DESIGN**

**Web design** encompasses many different skills and disciplines in the production and maintenance of [websites.](https://en.wikipedia.org/wiki/Website) The different areas of web design include web graphic design; [user interface](https://en.wikipedia.org/wiki/User_interface_design) design (UI design); authoring, including standardised code and [proprietary software;](https://en.wikipedia.org/wiki/Proprietary_software) [user](https://en.wikipedia.org/wiki/User_experience_design) experience design (UX design); and [search engine optimization.](https://en.wikipedia.org/wiki/Search_engine_optimization) Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term "web design" is normally used to describe the design process relating to the front-end (client side) design of a website including writing [markup.](https://en.wikipedia.org/wiki/Markup_language) Web design partially overlaps [web](https://en.wikipedia.org/wiki/Web_engineering) engineering in the broader scope of [web development.](https://en.wikipedia.org/wiki/Web_development) Web designers are expected to have an awareness of [usability](https://en.wikipedia.org/wiki/Web_usability) and be up to date with [web accessibility](https://en.wikipedia.org/wiki/Web_accessibility) guidelines.

Here’s why implementing these practices is important for your e-Commerce website:

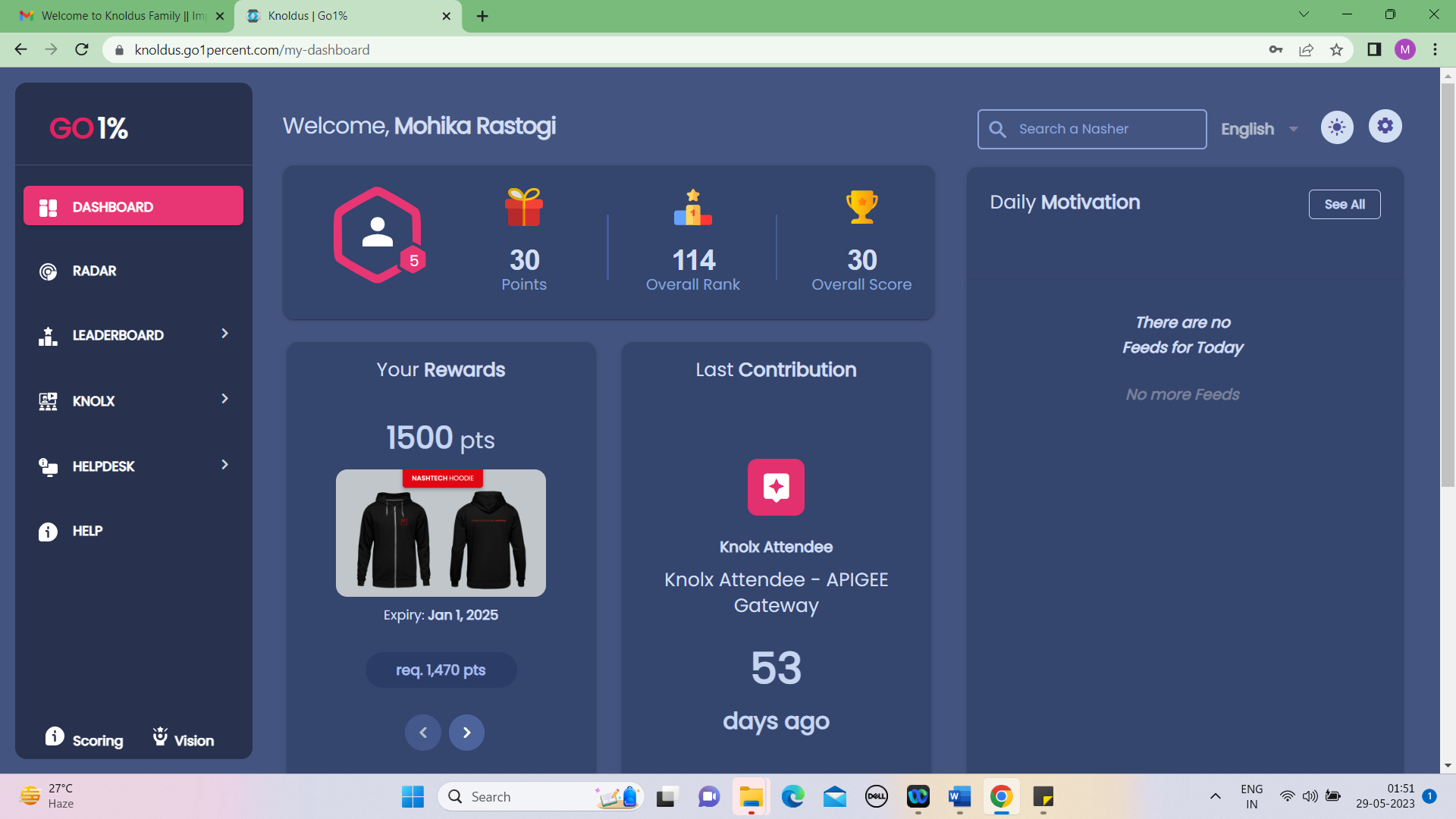
**To provide the best customer experience possible:** A great customer experience is about making it really simple for customers to buy products from your website, while also making it a great pleasure for them. These practices will help you create a website experience your customers will love.

**To boost sales through conversion rate optimisation:** Great design will help you convert more website visitors into paying customers, boosting your sales and enhancing business performance.

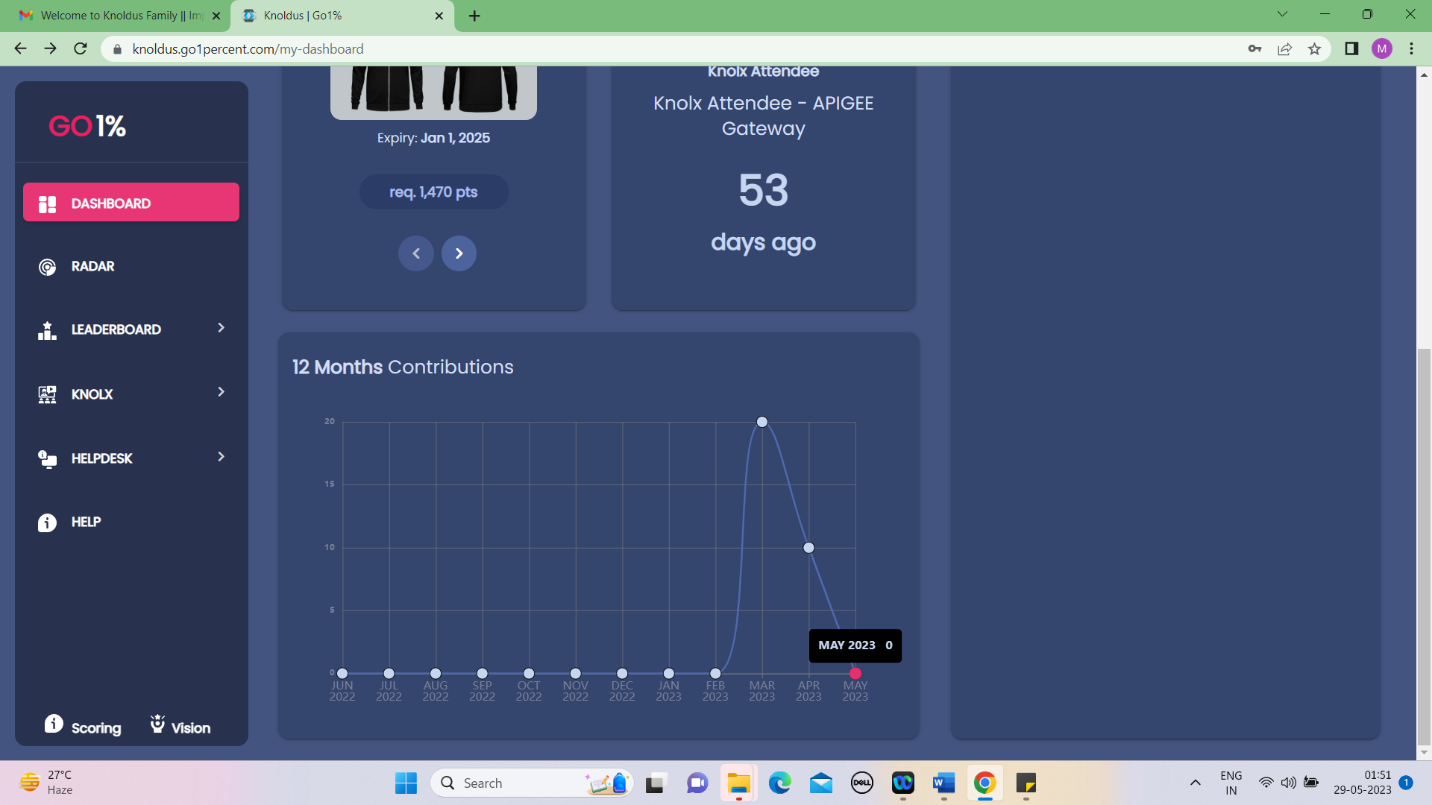
**To improve customer retention**: You don’t want website visitors or existing customers to leave your website and buy the same product from someone else. A great design will help you [grab the attention of your website visitors,](https://www.hostpapa.com/blog/web-design-development/keep-your-users-engaged-with-better-web-design/) and encourage your customers to be loyal towards your brand.

**To reinforce your brand:** Great design will speak volumes about who you are, what your brand stands for, and help you be perceived appropriately. At the end of the day, design is about great communication!

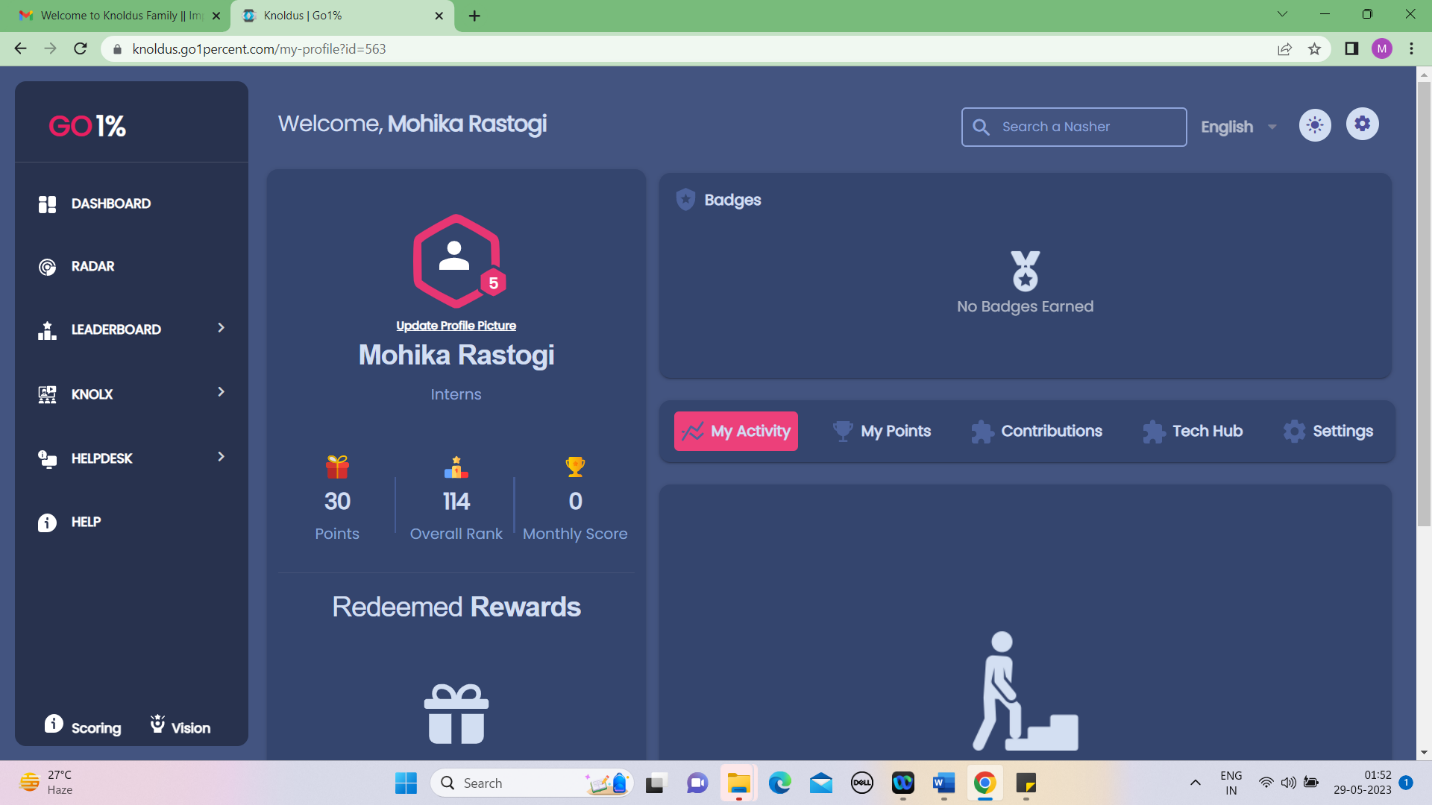
**To build customer relationships**: People tend to trust websites that are designed well, and therefore, will want to engage more with them. This helps with building great customer relationships with trust at its foundation.

****

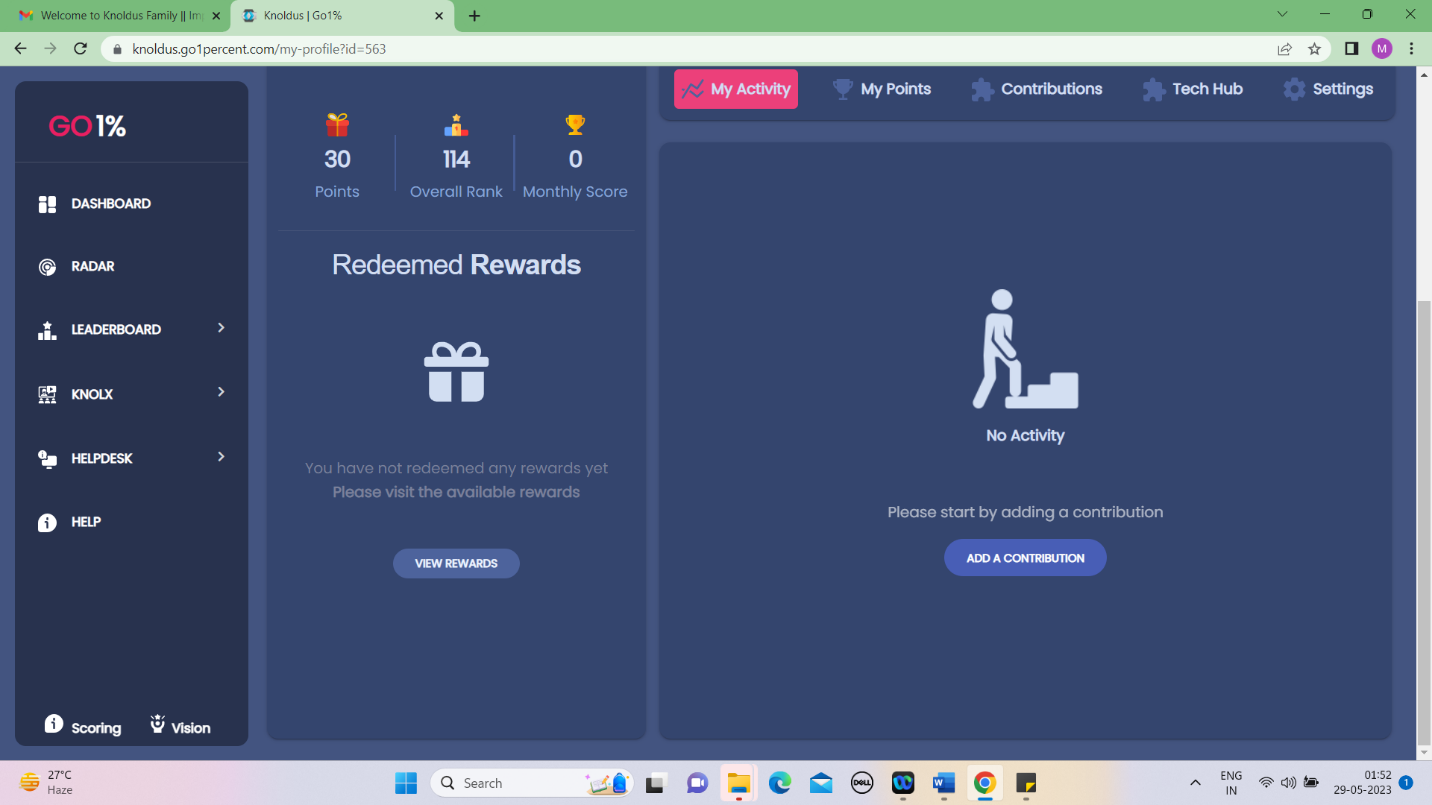
**Fig: 6.1 Dashboard**

****

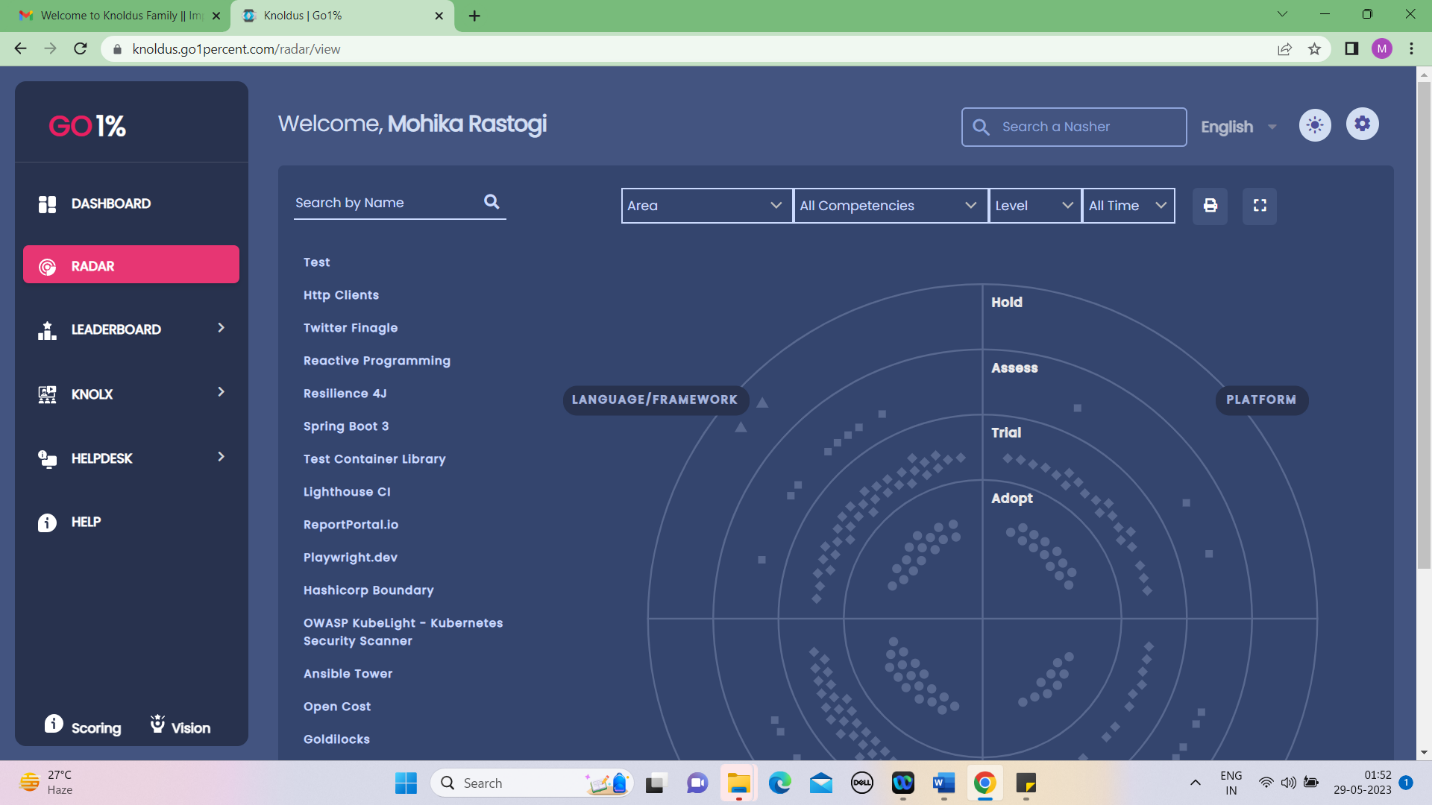
**Fig: 6.2 Monthly Contributions on Dashboard**

****

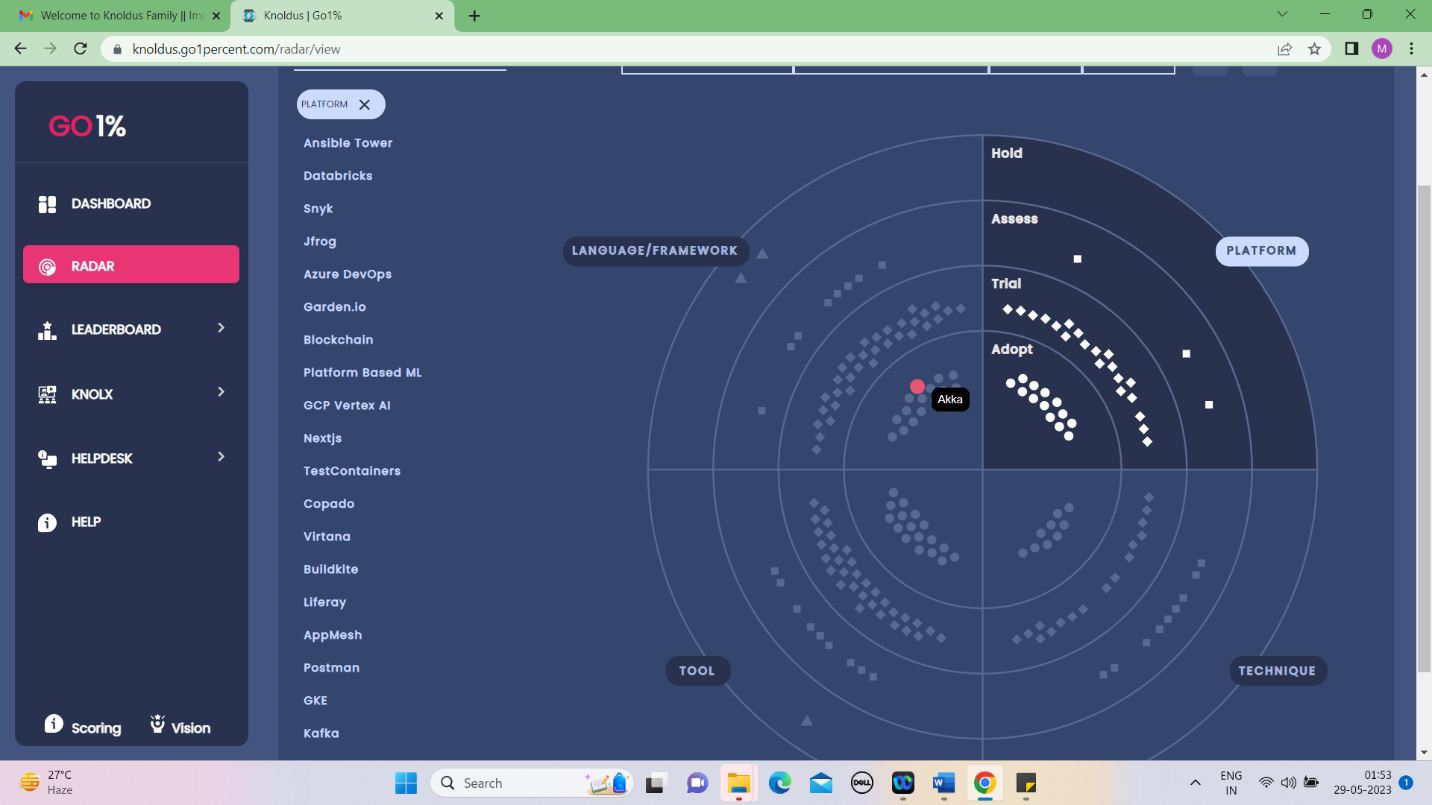
**Fig: 6.3 PROFILE**

****

**Fig: 6.4 Redeemed rewards**

****

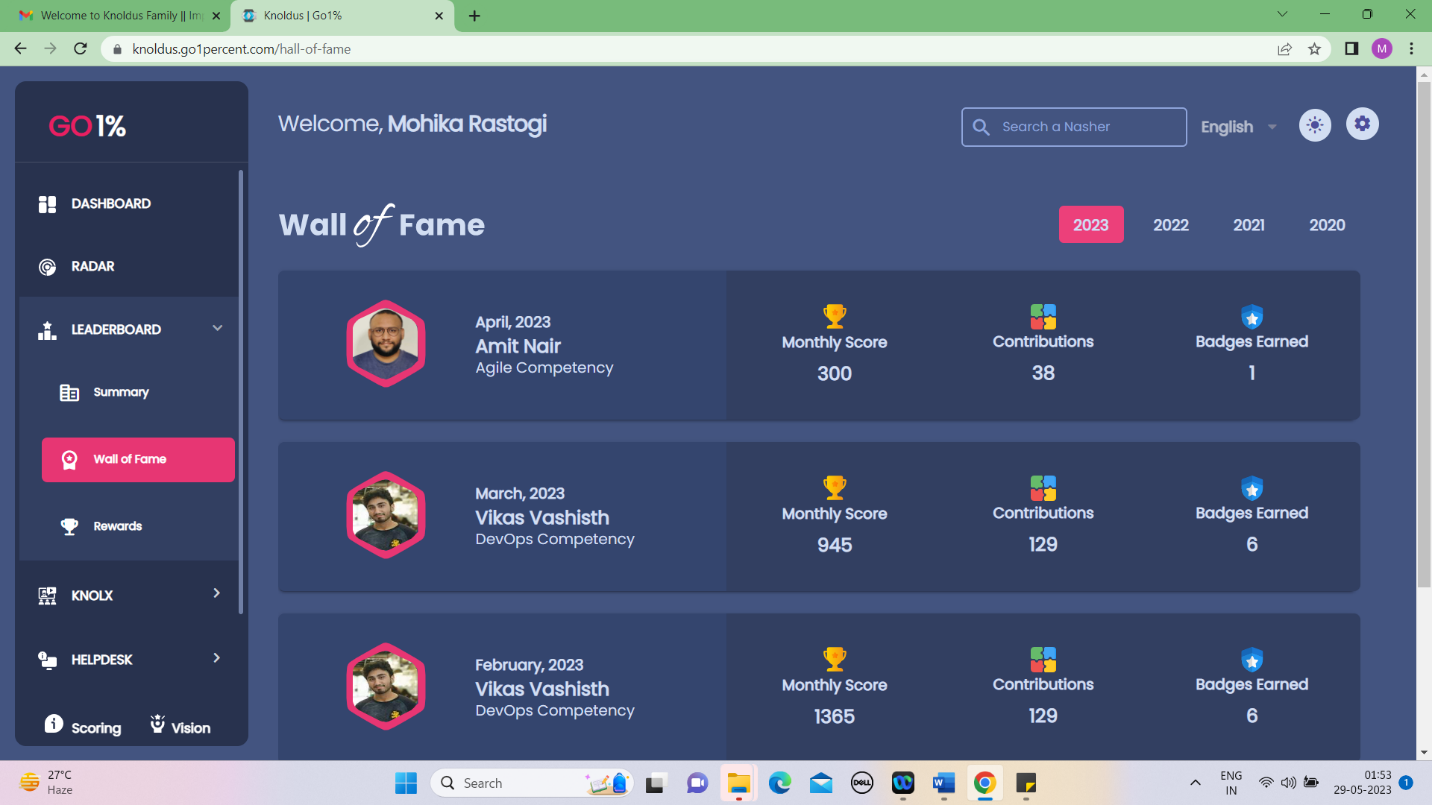
**Fig: 6.5 RADAR**

****

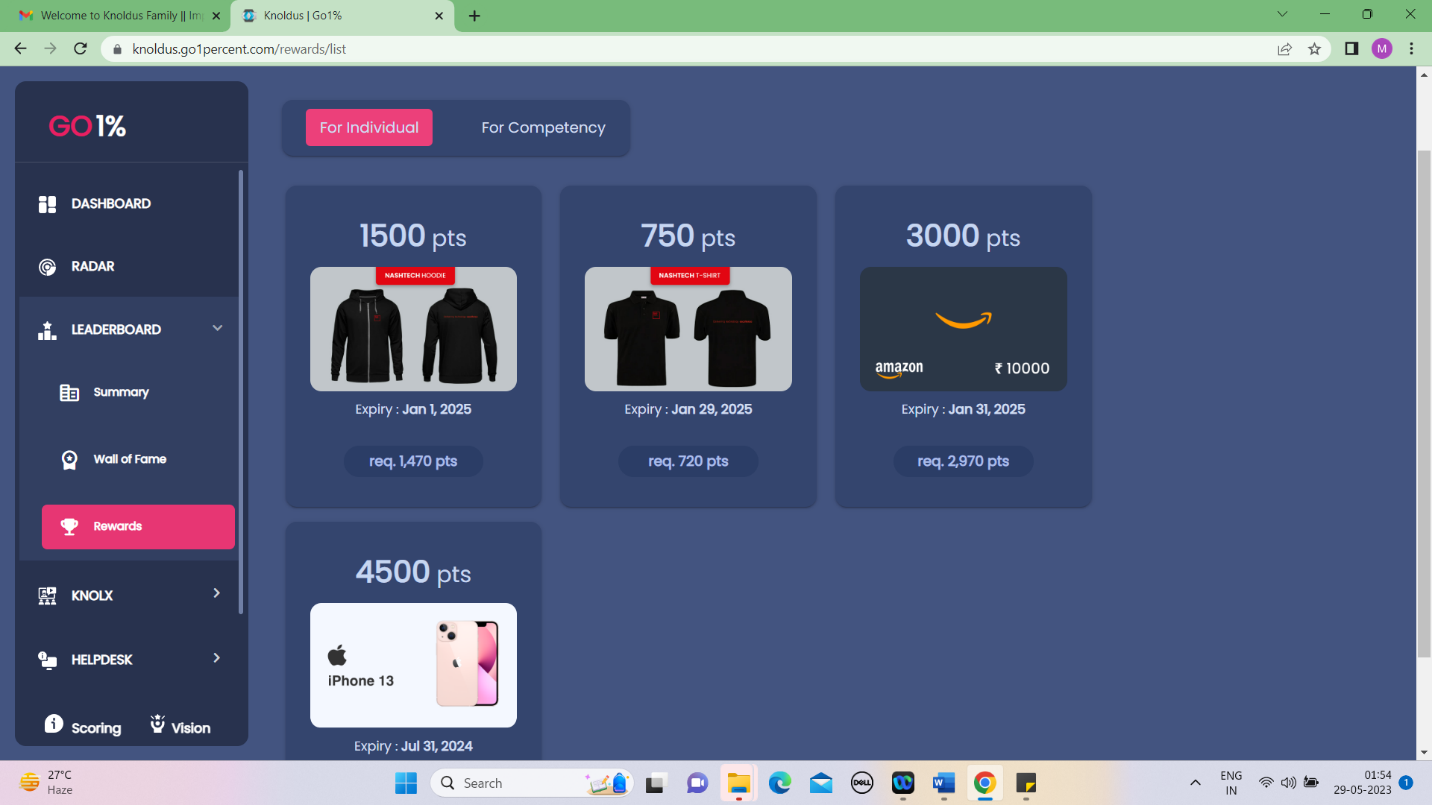
**Fig: 6.6 Track Learning status on RADAR**

****

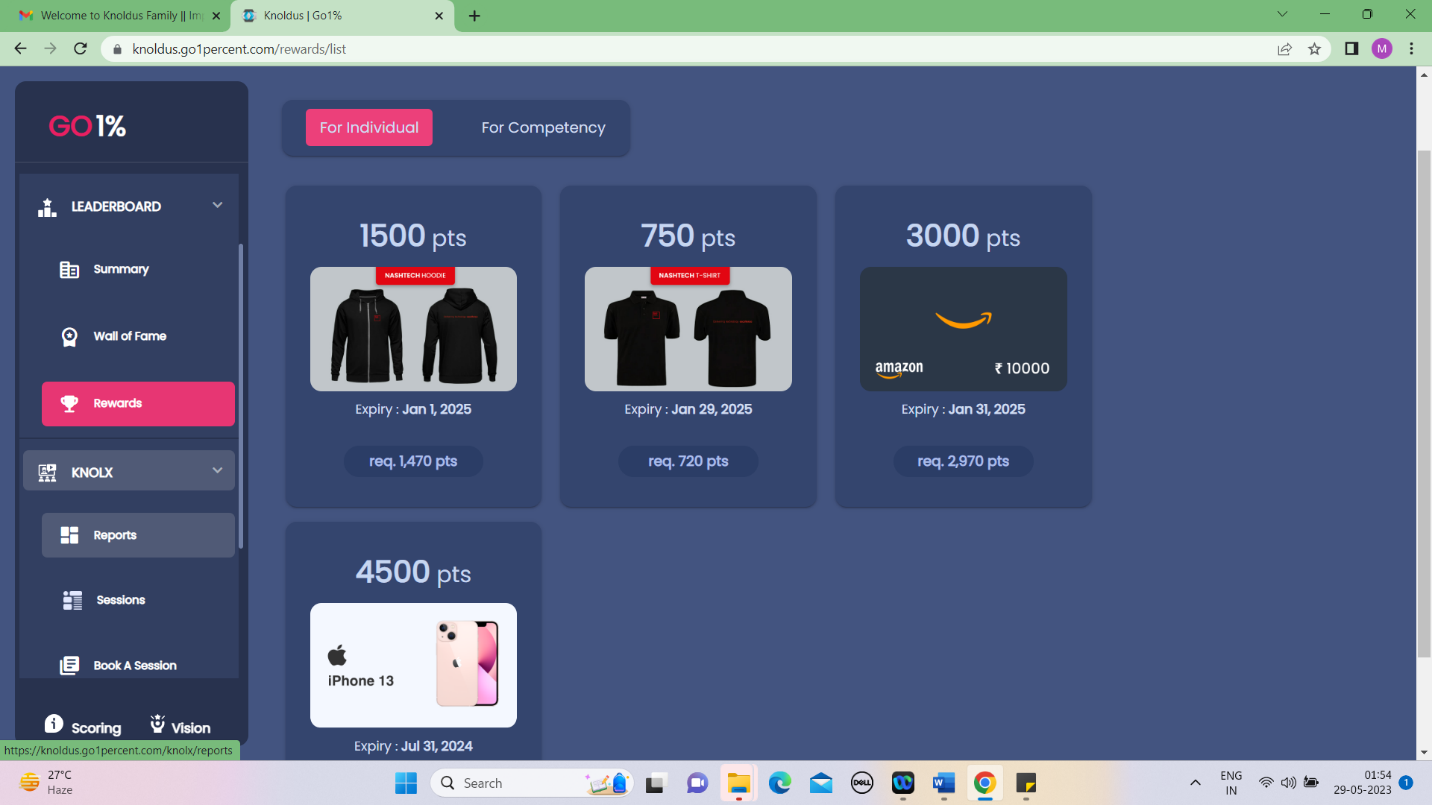
**Fig: 6.7 LEADERBOARD**

****

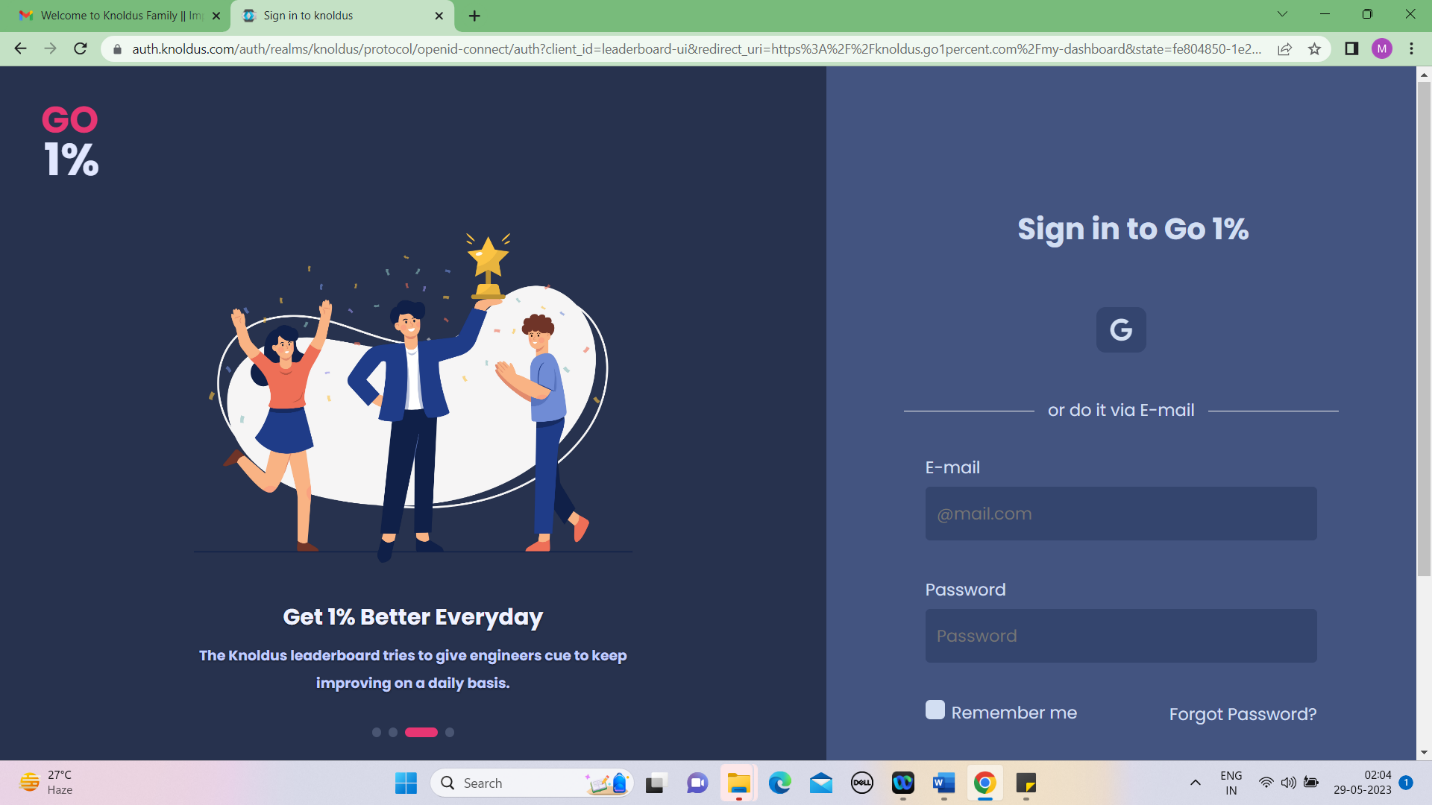
**Fig: 6.8 Wall of Fame on LEADERBOARD**

****

**Fig: 6.9 Rewards on LEADERBOARD**

****

**Fig: 6.10 KNOLX**

****

**Fig: 6.11 LOGIN/REGISTRATION**

**CHAPTER 7**

**TESTING**

**7.1 UNIT TESTING**

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.

A unit is a single testable part of a software system and tested during the development phase of the application software.

The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.

Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as **Unit testing** or **components testing**.

**Techniques of Unit Testing: -**

**7.1.1 White Box Testing**

White box testing techniques analyse the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

**7.1.2 Black Box Testing**

Black box testing is a technique of software testing which examines the functionality of software without peering into its internal structure or coding. The primary source of black box testing is a specification of requirements that is stated by the customer.

**7.1.3 Grey Box Testing**

Grey box testing is a software testing method to test the software application with partial knowledge of the internal working structure. It is a **combination of black box and white box testing** because it involves access to internal coding to design test cases as white box testing andtesting practices are done at functionality level as black box testing.

**7.2 INTEGRATION TESTING**

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Unit Testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.

**Techniques of Integrating Testing: -**

**7.2.1 Incremental Approach**

In the Incremental Approach, modules are added in ascending order one by one or according to need. The selected modules must be logically related. Generally, two or more than two modules are added and tested to determine the correctness of functions. The process continues until the successful testing of all the modules.

**7.2.2 Top-down Approach**

The top-down testing strategy deals with the process in which higher level modules are tested with lower-level modules until the successful completion of testing of all the modules. Major design flaws can be detected and fixed early because critical modules tested first. In this type of method, we will add the modules incrementally or one by one and check the data flow in the same order.

**7.2.3 Bottom – Up Approach**

The bottom to up testing strategy deals with the process in which lower-level modules are tested with higher level modules until the successful completion of testing of all the modules. Top level critical modules are tested at last, so it may cause a defect. Or we can say that we will be adding the modules from **bottom to the top** and check the data flow in the same order.

**7.2.4 Big Bang Approach**

In this approach, testing is done via the integration of all modules at once. It is convenient for small software systems, if used for large software systems identification of defects is difficult. Since this testing can be done after completion of all modules due to that testing team has less time for execution of this process so that internally linked interfaces and high-risk critical modules can be missed easily.

**7.3 SYSTEM TESTING**

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different type of tests with the purpose to exercise and examine the full working of an integrated software computer system against requirements.

To check the end-to-end flow of an application or the software as a user is known as **System testing**. In this, we navigate (go through) all the necessary modules of an application and check ifthe end features or the end business works fine, and test the product as a whole system.

It is **end-to-end testing** where the testing environment is similar to the production environment.

**Types of System Testing:**

**7.3.1 Performance Testing:**

Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.

**7.3.2 Load Testing:**

Load Testing is a type of software Testing which is carried out to determine the behaviour of a system or software product under extreme load.

**7.3.3 Stress Testing:**

Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.

**7.3.4 Scalability Testing:**

Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.

**7.4 ACCEPTANCE TESTING**

Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not. It is conducted as a kind of Black Box testing where the number of required users involved testing the acceptance level of the system. It is the fourth and last level of software testing.

User acceptance testing (UAT) is a type of testing, which is done by the customer before accepting the final product. Generally, UAT is done by the customer (domain expert) for their satisfaction, and check whether the application is working according to given business scenarios, real-time scenarios.

In this, we concentrate only on those features and scenarios which are regularly used by the customer or mostly user scenarios for the business or those scenarios which are used daily by the end-user or the customer.

**7.5 SOFTWARE VERIFICATION AND VALIDATION**

**7.5.1 Software Verification**

Verification testing includes different activities such as business requirements, system requirements, design review, and code walkthrough while developing a product.

It is also known as static testing, where we are ensuring that "**we are developing the right product or not**". And it also checks that the developed application fulfilling all the requirementsgiven by the client.

Verification is the process of checking that a software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfils the requirements that we have. Verification is **Static Testing**.

**Activities involved in verification:**

**7.5.1.1 Inspections:** Inspections involve a formal and systematic examination of documents, code, or other artifacts to identify defects, errors, or non-compliance with specified requirements.

**7.5.1.2 Reviews:** Reviews are a collaborative evaluation process where individuals examine documents, code, or other artifacts to identify potential issues, provide feedback, and assess compliance with specified requirements.

**7.5.1.3 Walkthroughs:** Walkthroughs involve a step-by-step examination of a system or process, where a presenter guides the participants through the details, functionality, and design to obtain feedback and identify potential issues

**7.5.1.4 Desk-checking:** Desk-checking is a methodical review process where an individual independently examines documents, code, or other artifacts without the need for formal meetings or presentations. It is often used to identify errors or omissions before more extensive verification activities take place.

**7.5.2 Software Validation**

Validation testing is testing where tester performed functional and non-functional testing.

Here **functional testing** includes [Unit Testing](https://www.javatpoint.com/unit-testing) (UT), [Integration Testing](https://www.javatpoint.com/integration-testing) (IT) and System Testing (ST), and **non-functional** testing includes User acceptance testing (UAT).

Validation testing is also known as dynamic testing, where we are ensuring that **"we have developed the product right."** And it also checks that the software meets the business needs ofthe client.

Validation is the process of checking whether the software product is up to the mark or in other words product has high level requirements. It is the process of checking the validation of product i.e., it checks what we are developing is the right product. it is validation of actual and expected product.

Validation is the **Dynamic Testing**.

**Activities involved in validation:**

**7.5.2.1 Black box testing:** Black box testing is a technique where the internal workings of the system or software being tested are not known or considered. Testers focus on validating the functionality, inputs, and outputs of the system without knowledge of its internal structure.

**7.5.2.2 White box testing:** White box testing is a technique where the internal structure, design, and implementation of the system or software being tested are known and considered. Testers have access to the code and use this knowledge to create tests that target specific paths, branches, or logic within the system.

**7.5.2.3 Unit testing:** Unit testing is a level of testing where individual units or components of the system, such as functions, modules, or classes, are tested in isolation. It focuses on verifying that each unit performs as expected and meets its specifications.

**7.5.2.4 Integration testing**: Integration testing is a level of testing where multiple units or components of the system are combined and tested together. It aims to uncover defects that may arise from the interaction between different units and ensure the integrated system functions correctly.

**7.6 TEST PROCEDURE**

A test procedure is a formal specification of test cases to be applied to one or more target program modules. Test procedures are executable. A process called the VERIFIER applies a test procedure to its target modules and produces an exception report indicating which test cases, if any, failed.

Test procedures facilitate thorough software testing by allowing individual modules or arbitrary groups of modules to be thoroughly tested outside the environment in which they will eventually reside. Test procedures are complete, self-contained, self-validating and execute automatically. Test procedures are a deliverable product of the software development process and are used for both initial checkout and subsequent regression testing of target program modifications.

Test procedures are coded in a new language called TPL (Test Procedure Language). The paper analyses current testing practices, describes the structure and design of test procedures and introduces the Fortran Test Procedure Language.

**7.7 TEST CASES**

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

* + 1. **Test Case: User Registration**
* **Description:** Verify that a new user can successfully register an account on the website.
* **Steps:**
* Navigate to the registration page.
* Fill in the required registration details.
* Submit the registration form.
* Verify that the user is redirected to the login page.
* Attempt to login using the newly registered credentials.
* Verify that the login is successful and the user is redirected to their dashboard.
  + 1. **Dashboard Progress Tracking**
* **Description:** Verify that the dashboard accurately tracks the user's progress and achievements.
* **Steps:**
* Login to the website as a registered user.
* Navigate to the dashboard.
* Complete various learning activities (e.g., attend a Knolx session, submit a blog post).
* Refresh the dashboard page.
* Verify that the progress indicators and achievements are updated to reflect the completed activities.
  + 1. **Test Case: Leaderboard Ranking**
* **Description:** Verify that the leaderboard accurately ranks users based on their contributions.
* **Steps:**
* Login to the website as multiple registered users.
* Engage in various learning activities (e.g., contribute to Knolx sessions, complete quizzes).
* Navigate to the leaderboard page.
* Verify that the users are ranked in the correct order based on their contributions.
* Verify that the user's own ranking is displayed prominently.
  + 1. **Test Case: Tech-hub Resource Access**
* **Description:** Verify that users can access and view learning resources in the Tech-hub module.
  + **Steps:**
* Login to the website as a registered user.
* Navigate to the Tech-hub module.
* Click on a specific learning resource (e.g., tutorial, course).
* Verify that the resource opens and can be viewed.
* Repeat the above steps for different types of resources.

**7.7.5 Test Case: Helpdesk Support Ticket Submission**

* **Description:** Verify that users can submit support tickets through the Helpdesk module.
* **Steps:**
* Login to the website as a registered user.
* Navigate to the Helpdesk module.
* Click on the "Submit a Ticket" button.
* Fill in the required details in the support ticket form.
* Submit the ticket.
* Verify that a confirmation message is displayed.
* Check the user's email for a notification regarding the submitted ticket.

**CHAPTER 8**

**CONCLUSION**

The Go 1% leaderboard platform offers a comprehensive and engaging solution for organizations to promote learning, development, and healthy competition among employees. The various modules, including the Dashboard, Radar, Leaderboard, Techhub, Knolx, Helpdesk, and Help, provide employees with a user-friendly interface and a wealth of features to enhance their personal and professional growth.

The Dashboard module serves as a central hub, allowing employees to track their progress, view their profile information, and monitor their achievements. By providing a graphical representation of their learning and development activities, employees can visualize the impact of their efforts, which serves as a strong motivator to continue their learning journey. The search bar feature also enables easy access to information about other employees within the organization, fostering collaboration and knowledge sharing.

The Radar module helps employees stay up-to-date with the latest technologies and tools, categorizing them into Adopt, Trial, Assess, and Hold categories. This empowers employees to make informed decisions about which technologies to incorporate into their work and personal projects, ensuring they stay aligned with industry best practices and advancements.

The Leaderboard module adds an element of healthy competition, allowing employees to compete with their peers and earn recognition for their achievements. With rankings, badges, and rewards, this module encourages employees to actively participate in learning and development activities, creating a culture of continuous improvement within the organization. This gamification approach can boost employee engagement, drive performance, and foster a sense of accomplishment.

The Techhub module serves as a repository of learning resources, providing employees with easy access to tutorials, courses, and documentation. This empowers employees to expand their knowledge and skills, keeping them updated with the latest industry trends and practices. The Knolx module further promotes knowledge sharing within the organization, enabling employees to share their expertise through presentations, webinars, and forums.

The Helpdesk and Help modules ensure that employees receive the necessary support and assistance throughout their learning journey. Whether it's technical issues or learning challenges, employees can rely on these modules to access relevant resources and guidance, ensuring a smooth and productive learning experience.

Overall, the Go 1% leaderboard platform offers a comprehensive solution for organizations to foster a culture of continuous learning and development. By leveraging the features and functionalities of this platform, organizations can empower their employees, enhance employee engagement, drive performance, and create a competitive edge. The gamification elements, collaborative features, and easy access to learning resources make it a valuable tool for organizations seeking to invest in their employees' growth and unlock their full potential. Implementing a similar platform can significantly benefit organizations by promoting a culture of continuous learning, fostering employee development, and ultimately driving organizational success.

**CHAPTER 9**

**RECOMMENDATION**

The Go1% Leaderboard project has the potential to significantly impact organizations by promoting a culture of continuous learning and improvement. Based on the analysis and evaluation of the project, the following recommendations are suggested for further enhancement and utilization of the leaderboard platform:

**9.1 Integration of Gamification Elements**

To further engage and motivate employees, consider incorporating additional gamification elements into the leaderboard platform. This can include features such as badges, achievements, and virtual rewards that employees can earn as they progress in their learning journey. Gamification can enhance the overall user experience and create a sense of competitiveness and fun.

**9.2 Collaboration and Social Learning Features**

Foster collaboration and social learning among employees by integrating features that facilitate knowledge sharing and interaction. This can include discussion forums, chat functionalities, and collaborative learning spaces where employees can exchange ideas, seek advice, and learn from each other's experiences. By encouraging collaboration, the platform can create a sense of community and promote a collective learning environment.

**9.3 Personalized Learning Recommendations**

Implement intelligent algorithms or machine learning techniques to provide personalized learning recommendations to employees based on their individual interests, goals, and learning history. By understanding each employee's unique learning preferences, the platform can suggest relevant courses, resources, and activities that align with their specific needs, enhancing the effectiveness of their learning experience.

**9.4 Integration with External Learning Resources**

Consider integrating the leaderboard platform with external learning resources, such as online courses, e-learning platforms, and knowledge repositories. This integration can provide employees with a wider range of learning opportunities and access to diverse learning materials. By offering a comprehensive collection of learning resources, the platform becomes a one-stop hub for employees to explore various learning avenues.

**9.5 Continuous Improvement and Feedback Mechanisms**

Establish a feedback mechanism within the platform that allows employees to provide feedback on the learning materials, courses, and overall user experience. Regularly analyse this feedback to identify areas for improvement and implement necessary enhancements. By actively seeking and incorporating user feedback, the platform can continuously evolve and adapt to meet the changing needs and preferences of the users.

**9.6 Mobile Accessibility**

Develop a mobile application or ensure the platform is fully responsive and accessible on mobile devices. With the increasing use of smartphones and tablets, providing a mobile-friendly experience allows employees to conveniently access the leaderboard platform anytime and anywhere. Mobile accessibility enhances engagement and encourages employees to utilize the platform even during their busy schedules.

**9.7 Data Analytics and Insights**

Utilize data analytics techniques to generate valuable insights and analytics reports for both employees and managers. These insights can include individual learning progress, overall team performance, trending topics, and skill gaps within the organization. By harnessing data-driven insights, managers can make informed decisions regarding training programs, skill development initiatives, and resource allocation.

**9.8 Scalability and Integration**

Ensure that the leaderboard platform is designed to scale and accommodate the growing needs of the organization. This includes considering future integrations with existing systems, such as human resources management systems (HRMS), learning management systems (LMS), and performance management systems. Seamless integration with other organizational platforms enhances the efficiency and effectiveness of the overall learning and development ecosystem.

In conclusion, by implementing these recommendations, the Go1% Leaderboard project can further empower organizations to foster a culture of continuous learning, engagement, and improvement. The incorporation of gamification elements, collaboration features, personalized recommendations, and mobile accessibility will enhance the user experience and drive employee participation. Additionally, leveraging data analytics and ensuring scalability and integration will enable organizations to make data-driven decisions and adapt the platform to their evolving needs. By embracing these recommendations, organizations can unlock the full potential of the leaderboard platform and maximize the impact of employee learning and development initiatives.

**CHAPTER 10**

**BIBLIOGRAPHY**

[1] Zhen Zhu, Shuai-fu Lin, Understanding entrepreneurial perceptions in the pursuit of emerging e-business opportunities: The dimensions and drivers, Computers in Human Behaviour, Volume-95(2019)

<(https://www.sciencedirect.com/science/article/pii/S0747563218300736>)

[2] Zhao Yingzi, Yu Zhaoji, Research on user trust evaluation of social e-commerce platform based on artificial neural network, Optik, Volume-271(2022)

<(https://www.sciencedirect.com/science/article/pii/S0030402622014516>)

1. Elena Higueras-Castillo, Francisco J. Liébana-Cabanillas, Ángel F. Villarejo-

Ramos, Intention to use e-commerce vs physical shopping. Difference between consumers in the post-COVID era, Journal of Business Research, Volume-157(2023)

<(https://www.sciencedirect.com/science/article/pii/S0148296322010876>)

1. Vinicius Facco Rodrigues, Lucas Micol Policarpo, Diórgenes Eugênio da

Silveira, Rodrigo da Rosa Righi, Cristiano André da Costa, Jorge Luis Victória Barbosa, Rodolfo Stoffel Antunes, Rodrigo Scorsatto, Tanuj Arcot, Fraud detection and prevention in e-commerce: A systematic literature review, Electronic Commerce Research and Applications, Volume-56(2022)

<(https://www.sciencedirect.com/science/article/pii/S1567422322000904>)

[5] Yang Bai, Hongxiu Li, Mapping the evolution of e-commerce research through co-word analysis: 2001–2020, Electronic Commerce Research and Applications, Volume-55(2022)

(<https://www.sciencedirect.com/science/article/pii/S1567422322000746>)

[6] Aras Alkis, Tekin Kose, Privacy concerns in consumer E-commerce activities and response to social media advertising: Empirical evidence from Europe, Computers in Human Behaviour, Volume-137(2022)

(<https://www.sciencedirect.com/science/article/pii/S0747563222002345>)

[7] Jie Wei, Zeling Xu, how does manufacturer’s self-operating channel interact with platform retailer’s E-commerce brand introduction? Transportation Research Part E: Logistics and Transportation Review, Volume-168 (2022)

(<https://www.sciencedirect.com/science/article/pii/S1366554522003283>)

[8] Xuan Liu, De Tong, Jiangming Huang, Wenfeng Zheng, Minghui Kong, Guohui Zhou, what matters in the e-commerce era? Modelling and mapping shop rents in Guangzhou, China, Land Use Policy, Volume-123(2022)

(<https://www.sciencedirect.com/science/article/pii/S0264837722004574>)

[9] Seema Baghla, Gaurav Gupta, Performance Evaluation of Various Classification Techniques for Customer Churn Prediction in E-commerce, Microprocessors and Microsystems, Volume-94(2022)

(<https://www.sciencedirect.com/science/article/pii/S0141933122002101>)

[10] Farid Shirazi, Nick Hajli, Julian Sims, Fred Lemke, The role of social factors in purchase journey in the social commerce era, Technological Forecasting and Social Change, Volume-183(2022)

(<https://www.sciencedirect.com/science/article/pii/S0040162522003857>)

[11] Mohammad Wasiq, Amar Johri, Prakash Singh, Factors affecting adoption and use of M-commerce services among the customers in Saudi Arabia, Heliyon, Volume 8, Issue 12(2022)

(<https://www.sciencedirect.com/science/article/pii/S2405844022038208>)

[12] Heleen Buldeo Rai, Laetitia Dablanc, Hunting for treasure: a systematic literature review on urban logistics and e-commerce data, Transport Reviews, 2022

(<https://www.sciencedirect.com/science/article/pii/S0144164722004536>)

[13] Jinchao Xu, Zihan Yang, Zi Wang, Jun Li, Xiaoshuan Zhang, Flexible sensing enabled packaging performance optimization system (FS-PPOS) for lamb loss reduction control in E-commerce supply chain, Food Control, Volume-145(2023)

(<https://www.sciencedirect.com/science/article/pii/S0956713522005874>)

[14] Odersky, M., Spoon, L., & Venners, B. (2016). Programming in Scala. Artima.

(<https://www.artima.com/pins1ed/>)

[15] Roestenburg, R., Bakker, R., & Williams, R. (2016). Akka in Action. Manning Publications.