

App Rationalization Tool Kit

S Karthick¹, Dr. V. Sathya²

¹Student, Department of Computer Applications, Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India.

²Assistant Professor, Department of Computer Applications, Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India.

To Cite this Article: S Karthick¹, Dr. V. Sathya², "App Rationalization Tool Kit", Indian Journal of Computer Science and Technology, Volume 03, Issue 02, May-August 2024, PP: 25-32.

Abstract: The main objective of this project entitled "App Rationalization Tool Kit" was developed to automate the manual process of the students. The App Rationalization Toolkit Project aims to streamline and optimize organizational app portfolios by developing a comprehensive framework for assessing, categorizing, and rationalizing existing applications. In today's rapidly evolving digital landscape, enterprises often find themselves burdened with a plethora of applications, many of which may be redundant, outdated, or underutilized. This project seeks to address these challenges by providing a structured methodology and set of tools to evaluate each application's value, alignment with business objectives, cost-effectiveness, and risk profile. By rationalizing the app portfolio, organizations can reduce complexity, improve efficiency, and allocate resources more strategically, ultimately enhancing their overall digital agility and competitiveness in the marketplace.

I.INTRODUCTION

The introduction of the App Rationalization Toolkit Project sets the stage by highlighting the increasingly complex landscape of enterprise applications and the challenges it poses for organizations. It underscores the importance of app rationalization as a strategic imperative in today's digital era, where businesses are constantly seeking to optimize their technology investments and align them with overarching business objectives.

The introduction outlines the overarching goals of the project, which include enhancing operational efficiency, reducing costs, mitigating risks, and fostering innovation. It also emphasizes the need for a systematic approach to app rationalization and previews the key components of the toolkit that will be developed as part of the project, setting the tone for the comprehensive methodology and tools that will follow in subsequent sections.

Ultimately, the introduction aims to captivate the reader's attention, convey the significance of the project, and create a sense of anticipation for the insights and solutions that will be unveiled in the rest of the document.

In today's dynamic business landscape, characterized by rapid technological advancements and evolving market demands, organizations face a multitude of challenges in managing their application portfolios. The proliferation of applications, driven by factors such as digital transformation initiatives, mergers and acquisitions, and decentralized decision-making, has led to complexity and inefficiency within many enterprises. As a result, IT departments often grapple with issues such as redundant systems, legacy applications, and underutilized resources, all of which contribute to increased costs, decreased agility, and heightened security risks.

II.MODULE

- Dashboard
- Create Clients
- Create Application List
- Create Stakeholders List
- Map Stakeholders List
- Creation of Template for Questionnaire
- Auto Assignment of Questionnaires

III.MODULE DESCRIPTION

Dashboard

Client Dash board

- Client-specific application and survey progress reports.
- Chart plotting for analysis.

Admin Dashboard

- Reports on the number of clients, templates, and applications.
- Client-wise progress reports, including updated, in-progress, and yet-to-start surveys.
- Chart plotting for analysis.

Create Clients

Capture essential client information, such as name, location, email, phone number, and creation status. Admin and clients can create and manage clients, with different creation statuses like "Hold," "Approved for 1 App," etc. The clients of the Dashboard module of the App Rationalization Toolkit Project encompass a diverse range of stakeholders within organizations seeking to streamline their application portfolios.

Create Application List

Define application details, including name, type of asset (hardware, network, software), and creation status. Admin and clients can create and manage application lists, with approval statuses similar to client creation. The module provides a centralized repository containing detailed information about all applications within the organization's portfolio, including names, vendors, versions, and relevant metadata.

Create Stakeholder List

Record stakeholder details, including name, role (hardware engineer, network engineer, etc.), and creation status. Admin and clients can create and manage stakeholder lists, including approval statuses. The Stakeholder List module of the App Rationalization Toolkit Project serves as a comprehensive repository of all individuals and groups who have a vested interest or play a role in the application rationalization process. This module captures key information about each stakeholder, including their role, responsibilities, contact details, and level of involvement in the project.

Map Stakeholders List

Establish connections between stakeholders and applications, indicating roles and approval/rejection status. Admin can approve mappings and update them, while clients can upload mappings. The module facilitates the identification and documentation of all stakeholders involved in the application rationalization process, including internal and external individuals or groups with a vested interest or influence. The module facilitates the identification and documentation of all stakeholders involved in the application rationalization process, including internal and external individuals or groups with a vested interest or influence.

Creation of Template for Questionnaires

Develop templates for questionnaires with statuses like "Approved" or "Rejected." Admin approval required, and clients can edit/add/delete inappropriate question. The creation of the Template for Questionnaires module within the App Rationalization Toolkit Project involves designing a structured framework to gather pertinent information from stakeholders and application owners. This template serves as a standardized set of questions and prompts aimed at capturing essential data points related to each application within the portfolio.

Auto Assignments of Questionnaires

Automatically assign questionnaires against applications for clients. Admin can view assignments, and clients can respond to questions based on templates and roles. Questionnaires are automatically assigned to application owners or designated subject matter experts responsible for the management and oversight of specific applications within the portfolio. Questionnaires are prioritized and assigned based on factors such as the criticality of the application, its strategic importance, or the urgency of the rationalization efforts, ensuring that high-priority applications are addressed promptly.

IV.SYSTEM REQUIREMENTS

Hardware Requirements**Minimum Requirement**

•Processor	:	Intel Core i5
•Speed	:	1(ghz) and Above
•RAM	:	8 GB and Above
•Monitor	:	15.6
•Input Devices	:	Keyboard, Touchpad

Software Requirement**Minimum Requirement**

• Frontend:	HTML, CSS and JAVASCRIPT
• Backend:	JAVA (SPRINGBOOT FRAMEWORK)
• Database:	MYSQL
• Web Server:	XAMPP
• Operating System:	Windows 10 and Above

V.TECHNOLOGY OVERVIEW

Hypertext Markup Language(Html)

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. 19

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages.

HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle bracket.

Tags such as `<div>` and `<p>` directly introduce content into the page. Other tags such as `<div class="header">` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. `<meta>` can be used to specify additional metadata about a document, such as its author, publication date, expiration date, language, page title, page description, keywords, or other information not provided through the other header elements and HTML attributes. Because of their generic nature, elements specify associative key-value pairs. In general, a meta element conveys hidden information about the document. Several meta tags can be used, all of which should be nested in the head element. The specific purpose of each elements.

In the App Rationalization Toolkit project, HTML (Hypertext Markup Language) is instrumental in shaping the user interface (UI) experience. HTML markup defines the structure of each web page, organizing elements like headers, footers, navigation bars, and content sections in a logical hierarchy. Through semantic HTML tags, such as `<h1>`, `<h2>`, and `<h3>`, the content is imbued with meaning, aiding accessibility and search engine optimization efforts. Additionally, HTML is utilized to craft various UI components, including buttons, forms, dropdown menus, and dialog boxes, which enable user interaction and data input. Moreover, HTML plays a pivotal role in facilitating responsive design principles, ensuring that the toolkit's interface adapts seamlessly to diverse devices and screen sizes. With its versatility and capability to integrate with other web technologies like CSS and JavaScript, HTML serves as the cornerstone for delivering an intuitive and user-friendly interface within the App Rationalization Toolkit, empowering stakeholders to efficiently manage their application portfolios.

JavaScript

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles. JavaScript runs on the client side of the web, which can be used to design programs on how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.

JavaScript can function as both a procedural and an object oriented language. JavaScript supports much of the structured programming syntax from C (e.g., if statements, while loops, switch statements, do while loops, etc.). One partial exception is scoping: JavaScript originally had only function scoping with `var`. ECMAScript 2015 added keywords `let` and `const` for block scoping, meaning JavaScript now has both function and block scoping. Like C, JavaScript makes a distinction between expressions and statements. One syntactic difference from C is automatic semicolon insertion, which allows the semicolons that would normally terminate statements to be omitted. Values are cast to numbers by casting to strings and then casting the strings to numbers. These processes can be modified by defining to `String` and `valueOf` functions on the prototype for string and number casting respectively.

In the App Rationalization Toolkit project, JavaScript serves as a cornerstone for enriching user interaction and functionality within the web-based application. Leveraging JavaScript's dynamic capabilities, developers enhance the toolkit's interactivity through features such as client-side validation, asynchronous data retrieval, and event-driven responses to user actions. JavaScript empowers the toolkit with the ability to manipulate the Document Object Model (DOM), enabling dynamic updates to content, structure, and styling in real-time. Furthermore, JavaScript facilitates seamless integration of third-party libraries and APIs, expanding the toolkit's functionality to include advanced features like data visualization, authentication, and interactive elements. With its crossbrowser compatibility and versatility, JavaScript ensures a responsive and engaging user experience, empowering stakeholders to effectively manage their application portfolios with efficiency and ease within the App Rationalization Toolkit.

Cascade Stylesheet(CSS)

CSS in the App Rationalization Toolkit project also serves as a powerful tool for customization and theming, offering users the ability to personalize their interface experience according to their preferences. Through CSS, users can modify color schemes, typography, and other visual elements to align with their organizational branding or individual tastes. This flexibility empowers users to create a workspace environment that reflects their unique style and enhances user satisfaction and engagement with the toolkit. Additionally, CSS plays a crucial role in optimizing the performance of the application, with developers employing techniques such as minification, bundling, and caching to streamline stylesheet delivery and reduce page load times.

By prioritizing performance considerations in CSS implementation, the toolkit ensures a smooth and efficient user experience, even under heavy usage or low bandwidth conditions. Furthermore, CSS is utilized in tandem with HTML and JavaScript to implement accessibility features, ensuring compliance with accessibility standards and providing an inclusive experience for all users, including those with disabilities. Through meticulous attention to CSS design and implementation, the App Rationalization Toolkit project aims to deliver a visually appealing, responsive, and accessible user interface that meets the diverse needs of its stakeholders. CSS (Cascading Style Sheets) is a fundamental component of web development, enabling the presentation and styling of HTML elements.

This powerful style language defines the visual layout, design, and formatting aspects of a website, allowing developers

to customize the appearance of text, images, backgrounds, and other 15 elements. With CSS, designers can control various properties such as color, font size, spacing, positioning, and responsiveness, ensuring consistency and enhancing user experience across different devices and screen sizes. Whether used independently or in conjunction with HTML and JavaScript, CSS empowers developers to create visually stunning and aesthetically pleasing websites, providing the flexibility and versatility needed to bring creative visions to life on the web.

Java

Java is a versatile and robust programming language widely used for developing a variety of software applications, ranging from web and mobile applications to enterprise-level systems. Known for its platform independence and object-oriented approach, Java allows developers to write code once and run it on any platform that supports Java without recompilation.

With its extensive standard library and rich ecosystem of third-party libraries and frameworks, Java facilitates rapid development and deployment of scalable, high-performance applications. Java's strong typing system and memory management features contribute to the language's reliability and stability, making it a preferred choice for building mission-critical applications in industries such as finance, healthcare, and telecommunications. Additionally, Java's support for multithreading enables developers to create concurrent and parallel applications, enhancing performance and responsiveness. Overall, Java's versatility, reliability, and scalability make it an indispensable tool for software development across various domains and industries.

Java's popularity stems from its comprehensive features, including its simplicity, security, and portability. The language's syntax is intuitive and easy to learn, making it accessible to both novice and experienced developers. Java's built-in security features, such as its robust sandbox environment and automatic memory management, help mitigate common vulnerabilities like buffer overflows and memory leaks, thus enhancing the overall safety of Java applications. Moreover, Java's platform independence allows developers to write code once and deploy it on multiple platforms without modification, thanks to the Java Virtual Machine (JVM), which abstracts away hardware-specific details. Java's suitability for large-scale projects is further underscored by its strong support for modular programming and code reusability. The language's modular architecture encourages the creation of modular, maintainable codebases, fostering collaboration and facilitating the management of complex software projects. Furthermore, Java's adherence to industry standards and best practices, combined with its active community and ongoing updates, ensures its continued relevance and longevity in the rapidly evolving landscape of software development.

In summary, Java's combination of simplicity, security, portability, and scalability makes it an indispensable tool for building a wide range of software applications, from simple utilities to enterprise-grade systems, across various platforms and industries.

In the App Rationalization Toolkit project, Java software plays a pivotal role in facilitating robust backend development and server-side processing. Leveraging Java's versatility and scalability, developers utilize the language to implement core functionalities such as data processing, business logic, and integration with databases and external systems.

Java servlets and frameworks like Spring Boot provide the foundation for creating scalable and high-performance server-side components, ensuring efficient handling of client requests and generation of dynamic responses. Furthermore, Java's extensive ecosystem of libraries and tools enables seamless data management operations, including retrieval, storage, manipulation, and analysis. By harnessing Java's capabilities, the App Rationalization Toolkit project aims to deliver a reliable, efficient, and feature-rich application backend that meets the diverse needs of its users.

Spring boot Framework

Spring Boot is a powerful framework for rapidly building enterprise-grade Java applications. It streamlines the development process by providing a comprehensive set of tools, conventions, and auto-configurations, allowing developers to focus on writing business logic rather than dealing with boilerplate code. With its embedded application server and easy-to-use configuration options, Spring Boot simplifies deployment and scalability, making it an ideal choice for microservices architectures and cloud-native applications. Additionally, Spring Boot integrates seamlessly with other Spring projects, such as Spring MVC and Spring Data, providing a cohesive ecosystem for building robust and scalable applications. Its extensive community support, extensive documentation, and active development ensure that developers have access to the latest features and best practices, making Spring Boot a preferred framework for building modern, scalable, and maintainable Java applications.

Spring Boot is renowned for its convention-over-configuration approach, which reduces the need for manual setup and configuration. This allows developers to get started quickly and focus on building business logic rather than spending time on infrastructure concerns. Furthermore, Spring Boot provides a variety of built-in features such as auto-configuration, which automatically configures the application based on dependencies present in the classpath, thus simplifying the setup process. Its robust dependency management capabilities ensure that developers can easily manage project dependencies and resolve conflicts, promoting a streamlined development experience.

In the App Rationalization Toolkit project, the Spring Boot framework serves as a cornerstone for building a robust and scalable backend infrastructure. Spring Boot simplifies development by providing a convention-over-configuration approach, significantly reducing boilerplate code and configuration overhead. Leveraging the powerful features of the Spring Framework, Spring Boot facilitates dependency injection and inversion of control, promoting modularity and testability while ensuring loose coupling between components. One of Spring Boot's key strengths lies in its autoconfiguration capability, which automates much of the setup process based on project dependencies and environment conditions, thereby streamlining development efforts. Additionally, Spring Boot's embedded server feature allows developers to package the application as a standalone executable JAR file, simplifying deployment and ensuring portability across different environments. With its comprehensive suite of features and robust ecosystem, Spring Boot empowers developers to build efficient, maintainable, and scalable backend systems for the App Rationalization Toolkit, enabling seamless management of application portfolios.

Furthermore, Spring Boot provides a cohesive ecosystem of tools and libraries that streamline various aspects of

application development. This includes Spring Data for simplifying data access and persistence through easy-to-use repositories and query methods, Spring Security for implementing robust authentication and authorization mechanisms to secure the application's resources, and Spring MVC for building RESTful APIs and handling web requests. Additionally, Spring Boot Actuator offers built-in monitoring and management endpoints, enabling administrators to monitor application health, gather metrics, and manage application lifecycle events easily. The framework's extensive support for testing, including integration with popular testing frameworks like JUnit and Mockito, ensures the reliability and stability of the application through comprehensive testing coverage. With its rich set of features, Spring Boot empowers developers in the App Rationalization Toolkit project to focus on building business logic and delivering value, while the framework handles the complexities of backend development.

XAMPP Server

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. The name XAMPP stands for cross-platform (X), Apache (A), MySQL (M), PHP (P), and Perl (P). It includes a collection of software components that are commonly used to develop dynamic web applications locally. The core components of XAMPP include the Apache HTTP Server, MySQL database server, PHP programming language, and Perl programming language.

XAMPP is designed to be easy to install and configure, making it an ideal solution for developers, students, and hobbyists looking to set up a local web development environment quickly. It provides a bundled package of server software that eliminates the need to install and configure each component individually. Additionally, XAMPP comes with a control panel that allows users to start, stop, and configure the various services included in the stack with ease. One of the key benefits of XAMPP is its portability, as it can run on multiple operating systems including Windows, macOS, and Linux. This makes it convenient for developers who need to work across different platforms or for users who want to carry their development environment with them on a portable storage device.

XAMPP is often used for developing and testing web applications locally before deploying them to a production server. It provides a sandbox environment where developers can experiment with different technologies, frameworks, and configurations without affecting live websites or data. Additionally, XAMPP includes support for additional programming languages and tools, making it a versatile platform for a wide range of web development projects.

Overall, XAMPP simplifies the process of setting up a local web server environment by providing an all-in-one solution that includes the necessary components for developing and testing web applications efficiently and effectively.

MYSQL

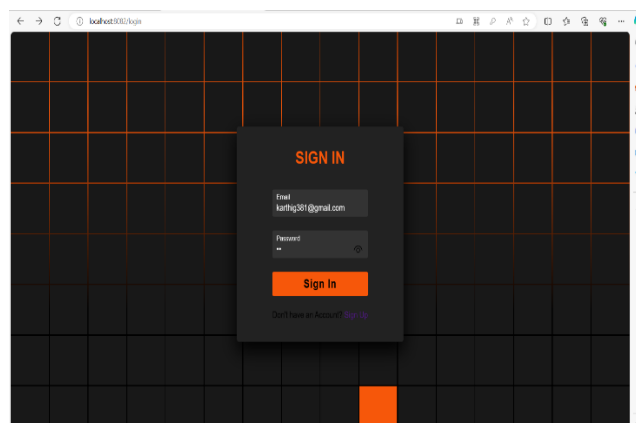
MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Wideness's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data to get on the database to control and the on if. This can be implemented to relational database to creation on it. 21 In our project, we harness the capabilities of Structured Query Language (SQL) to efficiently manage and interact with our database. 19 SQL acts as the foundation of our data management system, seamlessly organizing, retrieving, and manipulating information. By utilizing SQL queries, we are able to execute a range of operations including data insertion, record updates, and extraction of pertinent information for analysis.

Our database schema is meticulously designed to ensure data integrity and optimize performance, allowing us to meet the specific requirements of our project. Whether it involves handling extensive datasets or executing intricate transactions, SQL equips us with a resilient and adaptable toolkit to cater to our data-related needs. This significantly contributes to the overall success and functionality of our project. Moreover, SQL's reporting capabilities play a vital role in the analytical aspects of our project.

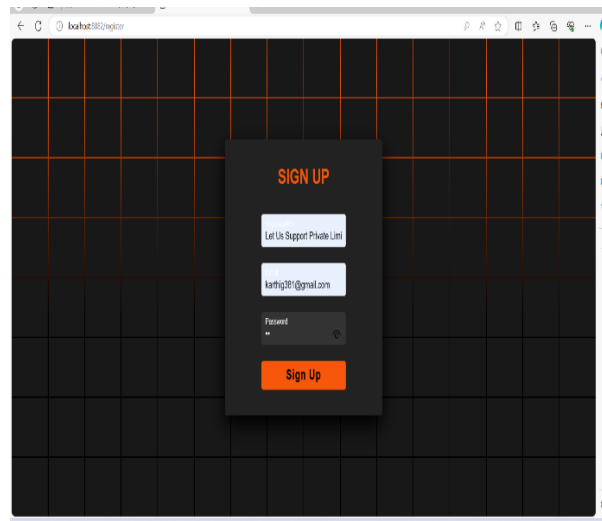
We leverage SQL queries to generate customized reports and extract valuable insights from our data, facilitating the decision-making process. The integration of business intelligence tools with SQL empowers our team to visualize data trends, create interactive dashboards, and conduct in-depth analysis, fostering a data-driven approach to problem-solving. To summarize, SQL plays a crucial role in managing the data for our project, providing a complete solution for organizing, securing, maintaining data integrity, and conducting analysis. The adaptability and powerful features of SQL greatly contribute to the overall effectiveness and productivity of our application, establishing SQL as an essential element in our development stack.

VI.SCRENN LAYOUTS

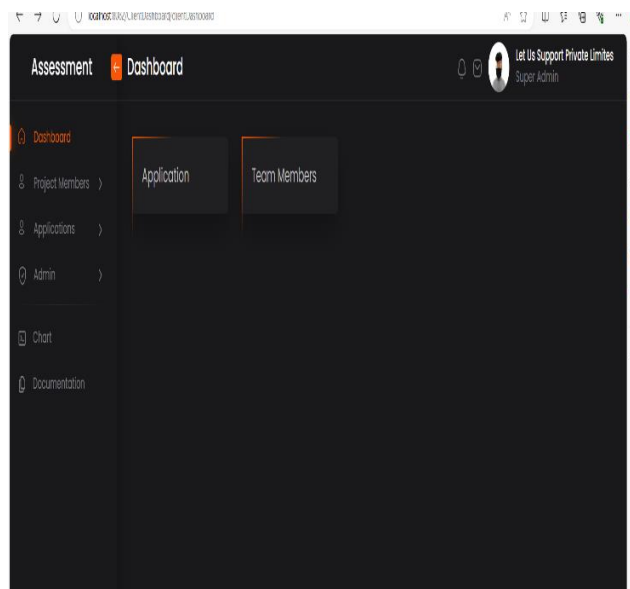
Admin And Client Login



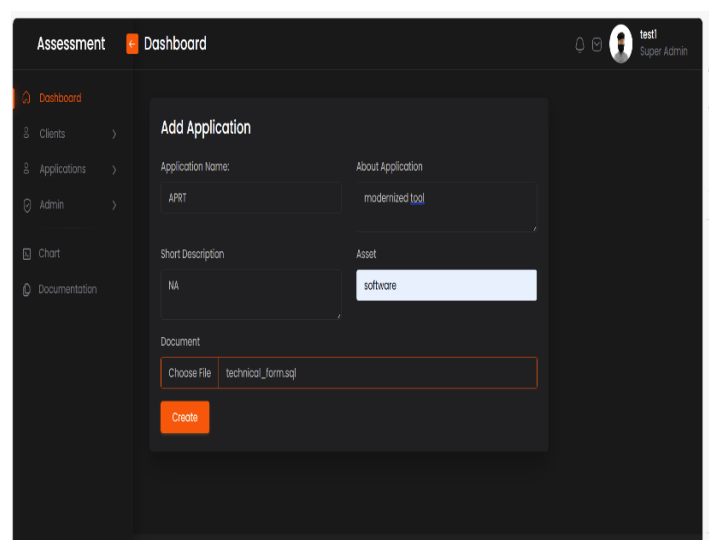
Client Register Page



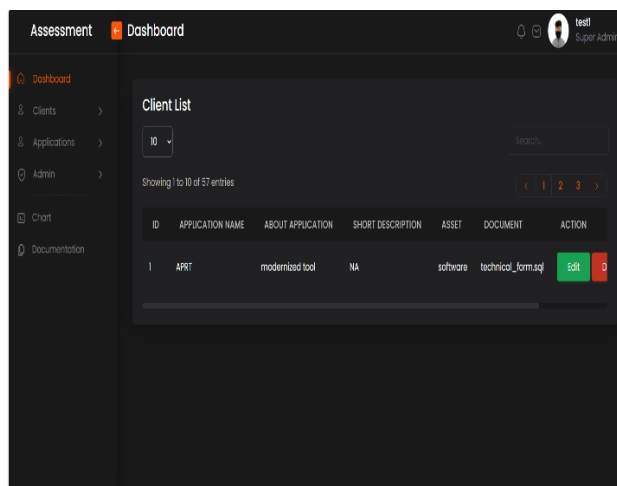
Client Dash Board



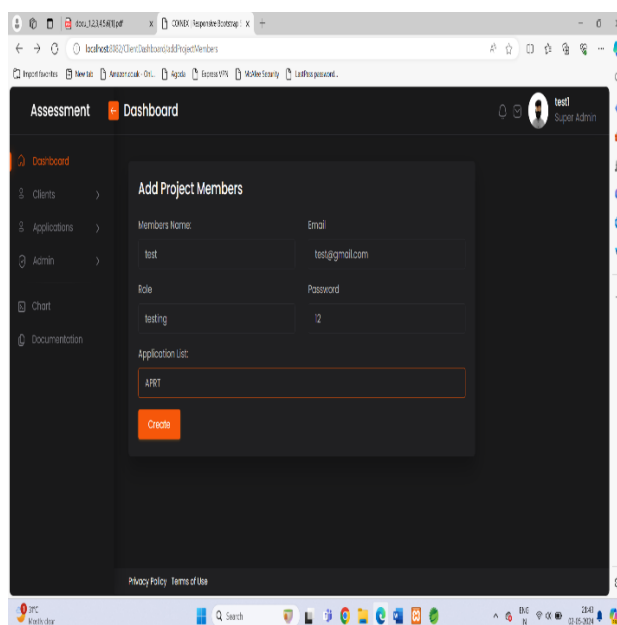
Add Application List



Application List



Add Project Member List



Project Member List

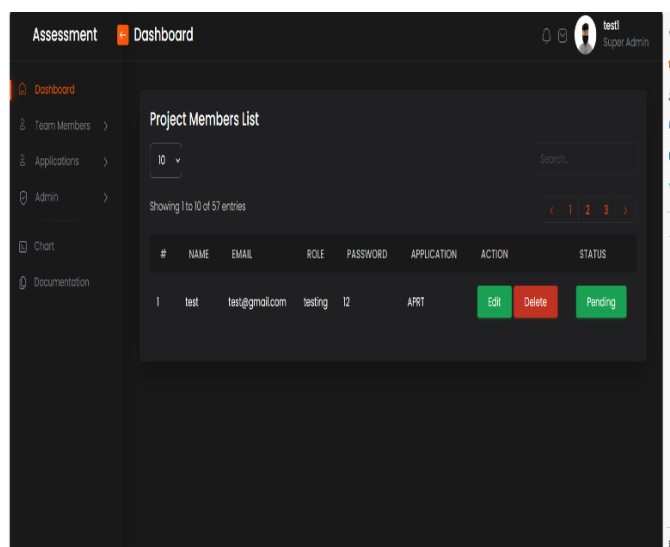
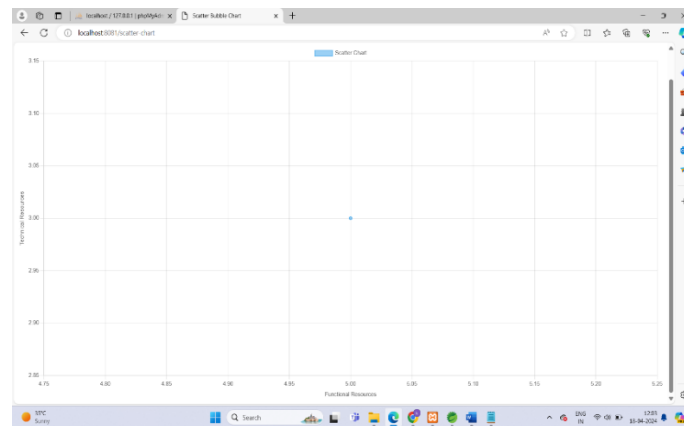


Chart Analysis



VII.CONCLUSION

After thorough analysis and implementation, the App Rationalization Toolkit project has reached its conclusion, marking a significant milestone in optimizing our digital ecosystem. Through diligent assessment and strategic decision-making, we have successfully rationalized our application portfolio, streamlining operations, reducing costs, and enhancing overall efficiency. By retiring redundant or outdated applications and consolidating overlapping functionalities, we have paved the way for a more agile and responsive IT infrastructure. Moreover, the toolkit developed during this project provides a sustainable framework for ongoing application governance, ensuring that our digital resources remain aligned with business objectives and responsive to evolving needs. As we move forward, we are well-positioned to leverage our optimized application landscape as a catalyst for innovation and growth.

Appendix

The appendix of the App Rationalization Tool Kit project serves as a comprehensive companion to the main report, offering a wealth of detailed documentation and supplementary materials. Within this section, stakeholders will find an exhaustive inventory of all applications utilized within the organization, complete with vital information such as vendor details, usage statistics, and version data. Furthermore, the appendix houses the assessment criteria utilized to evaluate each application's viability for retention, retirement, or consolidation, ensuring transparency and consistency in decision-making processes. Stakeholder interviews are encapsulated, providing valuable insights into user experiences, pain points, and requirements. Financial analyses, technical architecture diagrams, and risk assessments are meticulously documented, aiding in the formulation of informed strategies for application rationalization. Additionally, an implementation plan lays out the roadmap for executing recommended actions, emphasizing timelines, responsibilities, and necessary training. This comprehensive compilation of materials within the appendix enhances the depth and credibility of the project, facilitating clear understanding and effective decision-making for all involved parties.

Acknowledgment

Acknowledgment of the App Rationalization Tool Kit project is essential to recognize the collective effort and support that contributed to its successful completion. We extend our gratitude to all individuals and teams involved in the project, from stakeholders and subject matter experts to project managers and technical staff. Special thanks are due to [Name], whose expertise and guidance played a pivotal role in shaping the project's direction and outcomes. We also appreciate the cooperation and insights provided by various departments and personnel throughout the organization, without which this endeavor would not have been possible.

REFERENCES

- [1] Gartner. (2019). "Toolkit: Application Rationalization." Gartner Research. Retrieved from [insert link if available].
- [2] FREEMAN, E., ROBSON, E., & BATES, B. (2004). "HEAD FIRST DESIGN PATTERNS." O'REILLY MEDIA.
- [3] GAMMA, E., HELM, R., JOHNSON, R., & VLISSIDES, J. (1995). "DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE." ADDISON-WESLEY.
- [4] FOWLER, M. (2002). "PATTERNS OF ENTERPRISE APPLICATION ARCHITECTURE." ADDISON-WESLEY.
- [5] PRACTICES FOR TEAMS, PROGRAMS, AND THE ENTERPRISE." ADDISON-WESLEY.
- [6] CROCKFORD, D. (2008). "JAVASCRIPT: THE GOOD PARTS." O'REILLY MEDIA.