

Progress Tracking Dashboard using React.JS Review

Faizan Raza Sheikh¹, Hrishabh V. Petkar², Ketan Komawar³, Abdul Malik Sheikh⁴, Devashri Raich⁵

Final Year Students, Department of CSE^{1,2,3,4}

Guide, Department of CSE⁵

Rajiv Gandhi College of Engineering, Research & Technology, Chandrapur, India

faizanrazasheikh2002@gmail.comhrishabpetkar@gmail.comkomawarketan096@gmail.com

malikrs162@gmail.comdevashriraich@gmail.com

Abstract: *The Progress Tracking Dashboard built with React.js is a dynamic and user-friendly web application designed to empower users in monitoring, analyzing, and visualizing their progress across various domains. This dashboard harnesses the power of React.js, a popular JavaScript library for building interactive user interfaces, to provide an intuitive and responsive user experience. The Progress Tracking Dashboard built with React.js offers a robust solution for individuals and organizations seeking a versatile and efficient tool to monitor and analyze their progress in real-time. Leveraging React.js's capabilities, this dashboard provides an engaging and customizable user experience while ensuring data security and performance. Whether used for personal development, project management, or business analytics, this dashboard empowers users to make informed decisions and achieve their goals effectively.*

Keywords: Real-time Data Updates, Customizable Widgets, Visual Data Representation, User Authentication and Access Control, Responsive Design, Performance Optimization Integration Capabilities

I. INTRODUCTION

In the fast-paced world of today, where data-driven decision-making is crucial for individuals, businesses, and organizations, a Progress Tracking Dashboard is a valuable tool for monitoring, analyzing, and visualizing progress in various domains. These dashboards provide users with a comprehensive and real-time view of their performance, allowing for informed decision-making and goal achievement. One such innovative and dynamic solution is the Progress Tracking Dashboard, developed using the power of React.js.

React.js, a JavaScript library maintained by Facebook and a vibrant open-source community, has emerged as a leading choice for building interactive and responsive user interfaces. Its component-based architecture, efficient rendering, and virtual DOM (Document Object Model) make it an ideal framework for creating a dynamic and user-friendly Progress Tracking Dashboard.

This dashboard is designed to empower users with the tools and insights needed to stay on top of their goals and objectives. It offers a range of features, including real-time data updates, customizable widgets, interactive data visualizations, user authentication, and responsive design, all of which contribute to a seamless and intuitive user experience.

In this exploration of the Progress Tracking Dashboard using React.js, we will delve into the key features and functionalities that make it a standout solution. From real-time data tracking to secure user authentication, from insightful data visualization to customization options, this dashboard harnesses the capabilities of React.js to provide users with a powerful tool for progress tracking.

In the following sections, we will explore each of these features in detail, showcasing how React.js enables the creation of a versatile and efficient dashboard that meets the evolving needs of individuals and organizations alike. Whether it's personal development, project management, or business analytics, this Progress Tracking Dashboard is poised to be a game-changer, helping users achieve their goals and make data-driven decisions with ease.



Fig1 Progress Tracking Dashboard

II. SYSTEM REQUIREMENT

2.1 Software Requirements:

REACT.JS

React is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta and a community of individual developers and companies. React can be used to develop single-page, mobile, or server-rendered applications with frameworks like Next.js

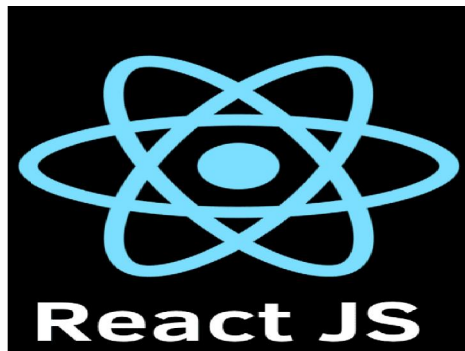


Fig 2 React JS

SYNCFUSION UI

It has many features like grouping, sorting, filtering, paging, swiping, multiple selection, drag and drop, and different layout types. The ListView control has been optimized to work with large amounts of data. The Scheduler is a fully featured calendar component that allows you to manage your time efficiently.



Fig 3 Syncfusion

TAILWIND CSS

Tailwind is a CSS framework that provides us with single-purpose utility classes which are opinionated for the most part, and which help us design our web pages from right inside our markup or .js/.



Fig4 Tailwind-CSS

REACT ROUTER

React Router is a **JavaScript framework that lets us handle client and server-side routing in React applications**. It enables the creation of single-page web or mobile apps that allow navigating without refreshing the page. It also allows us to use browser history features while preserving the right application view.



Fig 5 React Router

WEBPACK

Webpack will read from that input file, tracing through its dependency tree, combining all of the files together into a single file and outputting the file at the location you've specified as the output path. For this example, our input path is `src/main.js`, and we want to output the bundled file to `dist/bundle`.



Fig 6 Webpack

III. LIBRARY

1. SYNCFUSION/EJ2-REACT-CALENDARS

a) The `ej2-react-calendars` provides various features for handling date and time-related operations. This includes functionalities such as selecting dates, navigating through months and years, customizing date formats, and handling special dates.

b) It can also be used to customize the appearance and behavior of the calendar component according to their requirements. This includes options for changing themes, styling individual elements, defining special date templates, and configuring date ranges.

c) The `ej2-react-calendars` offers a robust solution for incorporating calendar functionalities into React applications, with a focus on features, customization options, accessibility, and ease of integration.

2. SYNCFUSION/EJ2-REACT-CHARTS :-

a) It offers a wide range of features for creating various types of charts, including line charts, bar charts, pie charts, area chart, and more.

b) `ej2-react-charts` supports interactive features that enhance user engagement and data exploration.

c) It supports various data-binding options, allowing developers to bind charts to local data arrays, remote data sources or data managers. This enables seamless integration of charts with real-time updates of chart data.

3. SYNCFUSION/EJ2-REACT-INPUTS :-

a) It is designed to provide a set of functionalities and customizable options to enhance the user experience and input interactions.

- b) It contain various input controls such as TextBox, NumericTextBox, AutoComplete, DropDownList, etc., which allow users to input different types of data.
- c) It contains a Built-in support for form validation, allowing developers to validate user input easily and provide feedback to users.

4. SYNCFUSION/EJ2-REACT-POPUPS :-

- a) ej2-react-popups is a collection of popup components designed for React applications. These components are versatile and provide various ways to display modal dialogs, popovers, and tooltips.
- b) Modal dialogs can be used for tasks such as displaying important messages, gathering user input, or confirming actions.
- c) Popovers are typically used to provide additional information or options related to the content the user is interacting with.
- d) Tooltips are small, floating boxes that appear when users hover over an element. They display helpful information or descriptions about the element being hovered over, providing users with context or guidance

IV. PROBLEM STATEMENT

Our organization is facing challenges in tracking and visualizing the progress of various projects, tasks, and team members efficiently. Currently, we rely on disparate tools and manual reporting, leading to delays in decision-making and a lack of real-time insights. To address this issue, we aim to develop a Progress Tracking Dashboard using React.js.

- **Real-time Visibility:** Create a dashboard that provides real-time visibility into project and task progress, allowing stakeholders to monitor status updates, timelines, and key performance indicators.
- **Data Integration:** Integrate data from various sources, such as project management tools (e.g., Jira, Trello), time tracking software, and other relevant systems to ensure a comprehensive overview.
- **Customization:** Enable users to customize their dashboard by selecting and arranging widgets or modules that are relevant to their roles and responsibilities.
- **User Authentication:** Implement secure user authentication and access control to ensure that only authorized personnel can view and interact with the dashboard.
- **Data Visualization:** Utilize data visualization techniques (e.g., charts, graphs, tables) to present complex project and task data in an understandable and actionable format
- **Alerts and Notifications:** Implement a notification system that alerts users to critical updates, deadlines, or issues that require immediate attention.
- **Mobile Responsiveness:** Ensure that the dashboard is responsive and accessible on various devices, including desktops, tablets, and smartphones.
- **Scalability:** Design the dashboard architecture to accommodate future growth and additional data sources, ensuring scalability and performance.
- **User Feedback:** Collect and incorporate user feedback to continuously improve the dashboard's usability and functionality.
- **Documentation and Training:** Provide comprehensive documentation and training materials to help users make the most of the Progress Tracking Dashboard.
- **Data Security:** Implement robust data security measures to protect sensitive project and task information from unauthorized access or breaches.

V. MODULES TO BE DEVELOPED

Developing a progress tracking dashboard using React.js requires creating various components and integrating them to provide a seamless user experience. Below are some key components and features you can consider implementing in your dashboard:

5.1. Progress Tracking:

- Utilize charts or graphs to visualize project progress over time.
- Show the completion status of tasks within each project.
- Display overall project completion percentages.



Fig7 Progress Tracking

NOTE:- This is going to be our first page of our dashboard in this weekend see our company monthly earnings yearly profit expenses budget refund how many orders we have get how many orders we have completed etc

5.2 User Authentication and Authorization

- Implement user authentication to secure access to the dashboard.
- Set up user roles and permissions for different levels of access (e.g., admin, team member).

5.3 Filtering and Sorting

- Allow users to filter projects and tasks by various criteria (e.g., status, priority, due date). -Provide sorting options to arrange projects and tasks by different attributes.

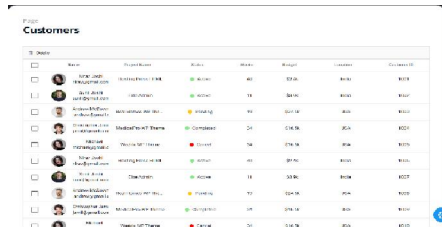


Fig 8 Filtering and Sorting

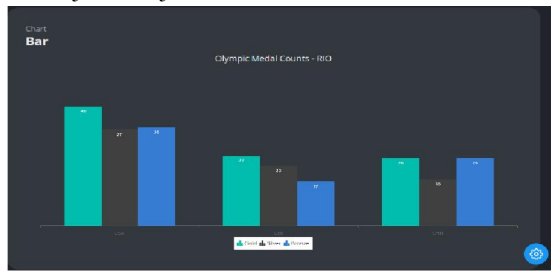
NOTE:- In this email we have shown that we can see our customer order customers refund customer pending amount shall be come from the customers and we can delete our customers if they have closed the bond with us.

5.4. Search Functionality:

- Implement a search bar to quickly find specific projects or tasks based on keywords.

5.5.Data Visualization:

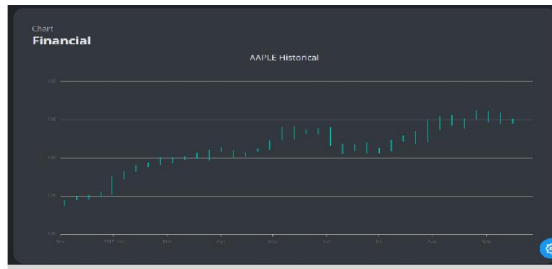
- Use React libraries like Chart.js or D3.js to create interactive data visualizations for project analytics>



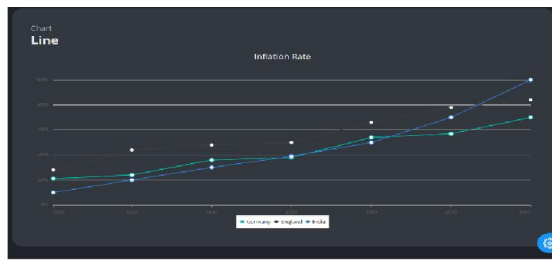
NOTE:- This is our bar data visualization in this we have shown a expenses and budgets of our company with the help of these are service providers or a customer can easily understand our earnings and expenses



NOTE:- This is our area chart in data visualization in this we have shown the area covered by our company for selling the product of our company the percentage shows how many area we have covered and in which area we have high amount of selling.



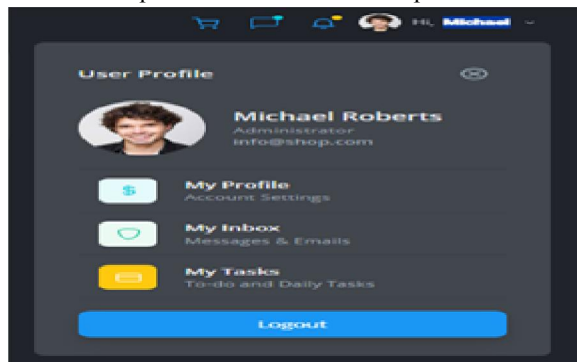
NOTE:- This is our financial chart in data visualization this is shown our financial economy of our company



NOTE:- This is our line chart in data visualization this shows the growth of our company in such a way as shown in a In a image it is easy to understand how our company is going on.

5.6. User Profile and Settings:

- Allow users to customize their profiles.
- Implement settings for notification preferences and dashboard preferences.



NOTE:- This is our user authentication area here we have put *MY PROFILE*, *MY INBOX*, *MY TASK* option and login and logout option this is shown that who is the authorized user of our dashboard

5. 7. API Integration:

- Connect your React dashboard to a backend API to fetch and update project and task data.

5. 8. Security Measures:

- Implement security best practices to protect user data and prevent common vulnerabilities.

5. 9. Performance Optimization:

- Optimize the performance of the dashboard by minimizing load times and optimizing API requests

VI. CONCLUSION

In conclusion, creating a project progress dashboard is a valuable practice for effectively tracking your progress and ensuring project success. By understanding the importance of a dashboard, including key elements, following the steps to create and maintain it, and implementing tips for maximizing its use, you can stay organized, informed, and in control throughout your project journey.

REFERENCES

- [1]. Creating a Dashboard to Increase Efficiency in tracking and Reporting on Research Data Requests, Abigail Nerogic, Virginia Lawson, Larisa Rodgers, Wake Forest School of Medicine
- [2]. Impact of a Patient Tracking Dashboard on the Timeliness for Chart Check in a Community-Based Radiation Oncology Department, G. ThorntonP. BeickmanK. GrassmanJ.A. Masino
- [3]. .Developing a dashboard for program evaluation in competency-based training programs: a design-based research project, RobertoMartinez-MaldonadoMonashUniversity(Australia)Yi-ShanTsaiMonashUniversity, (Australia) Namrata Srivastava Monash University (Australia)
- [4]. Developing the VA Geriatric Scholars Programs' Clinical Dashboards Using the PDSA Framework for Quality Improvement, Zachary Burningham^{1,2} Regina Richter Lagha³Brittany Duford-Hutchinson²Carol Callaway-Lane, Brian C. Sauer^{1,2} Ahmad S. Halwani^{2,5} Jamie Bell^{1,2} Tina Huynh^{1,2} Joseph R. Douglas, B. Josea Kra