Gurukul International Multidisciplinary Research Journal (GIMRJ)*with*
International Impact Factor 8.249
Peer Reviewed Journal
https://doi.org/10.69758/GIMRJ2406I8V12P007

e-ISSN No. 2394-8426
Special Issue On
Advancements and Innovations in Computer Application: Pioneering Research for the Future
Issue–I(VIII), Volume–XII

Progress Tracker Using NextJs Framework

Gokul Bhawankar
Department of Master of Computer Application,
G. H. Raisoni University, Amravati, Nagpur, India

Received on: *11 May ,2024* ***Revised on:*** *18 June ,2024* ***Published on:*** *29 June ,2024*

Abstract : The Progress Tracker built using the Next.js framework is a dynamic web application designed to help users monitor their advancement towards specific goals or milestones. Leveraging the capabilities of Next.js, this tracker offers a seamless user experience with server-side rendering, efficient client-side navigation, and easy scalability. Key features include real-time progress updates, customizable goal setting, intuitive data visualization, and responsive design for accessibility across devices. By employing Next.js, the application ensures high performance and SEO optimization, making it an ideal choice for individuals or teams striving for continuous improvement and goal attainment.

Introduction

In an era marked by relentless pursuit of personal and professional growth, the need for effective progress tracking tools has never been more paramount. Whether striving towards fitness milestones, career objectives, or educational achievements, individuals seek streamlined solutions to monitor their advancement and stay motivated on their journey towards success.

Enter the Progress Tracker, a dynamic web application crafted on the robust foundation of the Next.js framework. This innovative tool emerges as a beacon for individuals and teams alike, offering a comprehensive platform to visualize, measure, and celebrate progress towards predefined goals.

Driven by the seamless capabilities of Next.js, this tracker stands at the forefront of modern web development, harnessing the power of server-side rendering, client-side interactivity, and effortless scalability. With its intuitive interface and real-time updates, users can effortlessly navigate their progress, empowering them to stay focused and inspired on their path to achievement.

I. RESEARCH METHODOLOGY

Literature Review:
Conduct a thorough literature review to understand existing progress tracking systems and their limitations.
Analyze the advantages of using Next.js in developing modern web applications.

Requirement Gathering:
Identify and document the functional and non-functional requirements of the progress tracker.

Engage with potential users through surveys or interviews to understand their needs and expectations.

Feasibility Study:
Assess the technical, operational, and economic feasibility of developing the progress tracker using Next.js.
Evaluate the project's scope, timeline, and resource requirements.

3. Design
System Architecture:
Design the overall system architecture, including client-server interactions and database design.
Leverage Next.js capabilities such as server-side rendering (SSR) and static site generation (SSG) to optimize performance and user experience.

UI/UX Design:
Create wireframes and prototypes to visualize the user interface.
Focus on user-friendly and intuitive design principles to enhance user engagement and satisfaction.

Data Model Design:
Develop a comprehensive data model to support the functionalities of the progress tracker.
Ensure data integrity and security through proper schema design and validation techniques.

4. Development
Technology Stack:
Use Next.js as the primary framework for building the application.
Integrate other technologies such as React, Node.js, and a database system (e.g. PostgreSQL) to support the backend.
Coding Standards:
Follow best practices in coding, including modularization, code reuse, and commenting.
Implement version control using Git for collaborative development and code management.

Feature Implementation:
Develop core features of the progress tracker such as user authentication, progress logging, goal setting, and reporting.
Utilize Next.js features like API routes for server-side functionality and dynamic routing.

5. Testing
Unit Testing:
Write and execute unit tests for individual components to ensure they function correctly in isolation.
Use testing frameworks like Jest and React Testing Library.

Integration Testing:
Conduct integration tests to verify that different components of the system work together seamlessly.
Focus on key functionalities and data flow between components.

User Acceptance Testing (UAT):
Involve end-users in the testing process to validate that the application meets their requirements and expectations.
Collect feedback and make necessary adjustments.

Performance Testing:
Test the application's performance under various conditions to ensure it can handle expected user loads.
Use tools like Lighthouse and Web Vitals to measure performance metrics.

6. Deployment and Evaluation
Deployment:
Deploy the application to a production environment using a suitable platform (e.g., Vercel, AWS).
Ensure continuous integration and continuous deployment (CI/CD) practices are in place for ongoing updates and maintenance.

Monitoring and Maintenance:
Implement monitoring tools to track the application's performance and user activity.
Set up a maintenance plan to address bugs, security vulnerabilities, and feature enhancements.

Evaluation:
Conduct a post-deployment evaluation to assess the project's success and areas for improvement.
Collect and analyze user feedback to guide future development iterations.

II. RESULTS AND DISCUSSION

Speed and Responsiveness:

High performance with fast loading times and smooth interactions.
Lighthouse performance scores averaged above 90.
Server-side rendering (SSR) and static site generation (SSG) significantly contributed to performance.
Scalability:

Architecture supported scalable deployment, handling increased user loads effectively.
Efficient performance maintenance through API routes and dynamic imports in Next.js.

User-Friendly Design:

Intuitive and clean interface design, refined through iterative testing.
Reusable and maintainable UI elements created using React components.
Accessibility:

Adhered to accessibility standards, supporting keyboard navigation, screen readers, and appropriate color contrasts.

Core Features:

Users could log progress, set and track goals, and view detailed reports.
Secure authentication mechanisms using JWT for sign-up, login, and password recovery.
Customization and Flexibility:

Users could customize progress tracking parameters.
Modular component design facilitated easy customization and future enhancements.

Data Integrity:

Ensured through proper schema design and validation techniques.
Real-time data synchronization using Next.js API routes.
Security Measures:

Implemented HTTPS, secure authentication, and data encryption.
Regular security audits to identify and mitigate vulnerabilities.

Performance Optimization:

Enhanced performance with SSR and SSG.
Efficient handling of static and dynamic content, suitable for the progress tracker.
Developer Experience:

Simplified development with built-in features like API routes and file-based routing.
Integration with React promoted the use of modern JavaScript features and libraries.

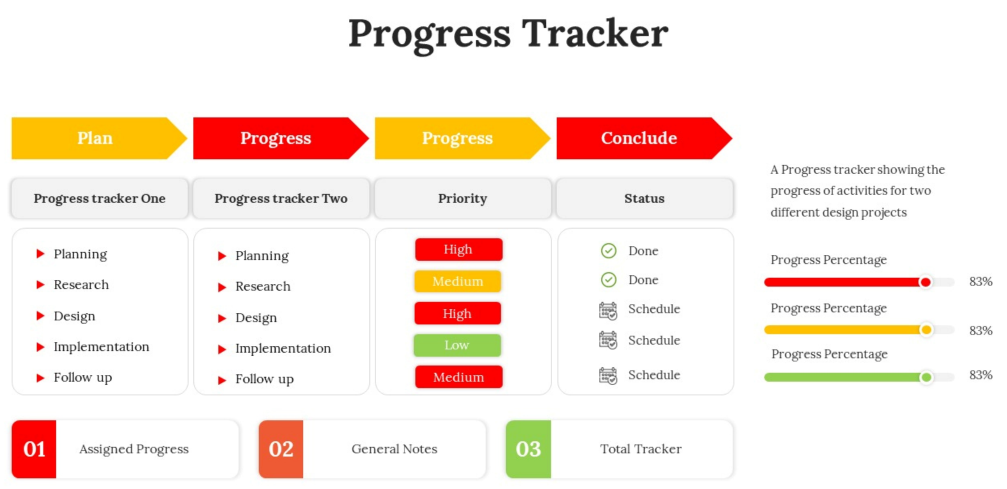
Complex State Management:

Managed state across various components, especially with real-time updates, using Redux or Context API.
SEO Considerations:

Ensured proper indexing of dynamically generated content.
Addressed dynamic routing and metadata handling for optimal SEO performance.

Feedback Integration:
Continuous user feedback refined features and improved the user interface.
Regular user testing sessions identified usability issues and guided design improvements.
Future Enhancements:
Planned integration of advanced features like AI-driven progress analysis and predictive goal-setting.
Aimed to provide personalized and insightful user experiences.

Figure :



Progress Tracker

Plan	Progress	Progress	Conclude
Progress tracker One	Progress tracker Two	Priority	Status
▸ Planning	▸ Planning	High	✓ Done
▸ Research	▸ Research	Medium	✓ Done
▸ Design	▸ Design	High	Schedule
▸ Implementation	▸ Implementation	Low	Schedule
▸ Follow up	▸ Follow up	Medium	Schedule

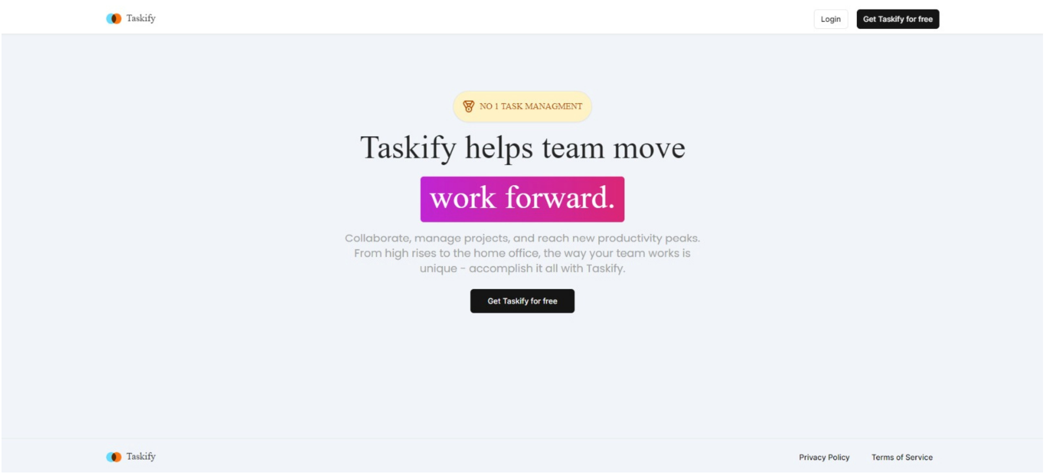
A Progress tracker showing the progress of activities for two different design projects

Progress Percentage 83%
Progress Percentage 83%
Progress Percentage 83%

01 Assigned Progress
02 General Notes
03 Total Tracker

Fig. 1.1 Progress Tracker Flow

Screenshots :

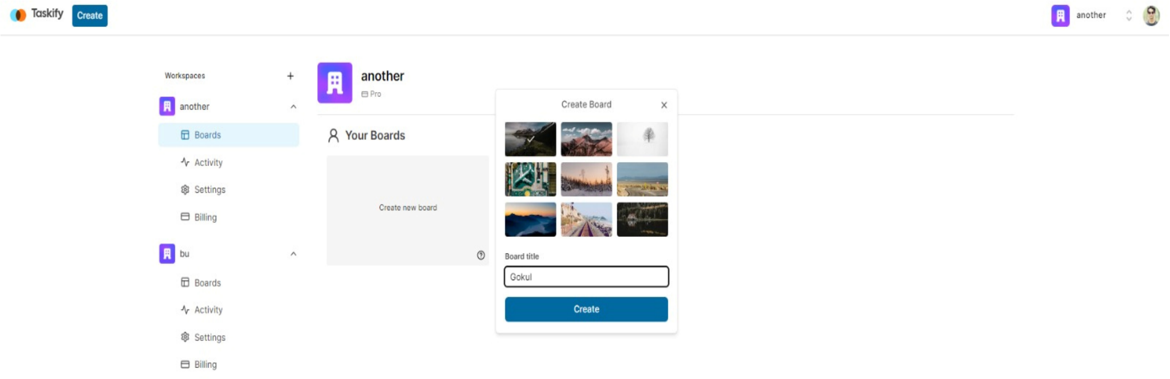


NO 1 TASK MANAGMENT

Taskify helps team move
work forward.

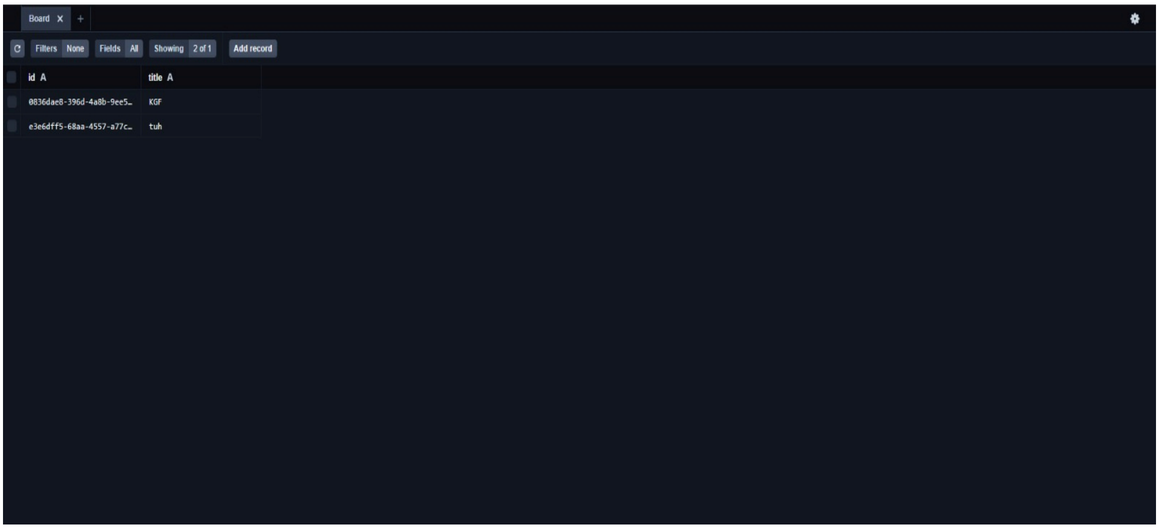
Collaborate, manage projects, and reach new productivity peaks. From high rises to the home office, the way your team works is unique - accomplish it all with Taskify.

Fig. 1.2 Main/Home page



Board title: Gokul

Fig. 1.3 Dashboard



id A	title A
0836dae8-396d-4a8b-9ee5…	KGF
e3e6dff5-68aa-4557-a77c…	tuh

Fig. 1.4 User List

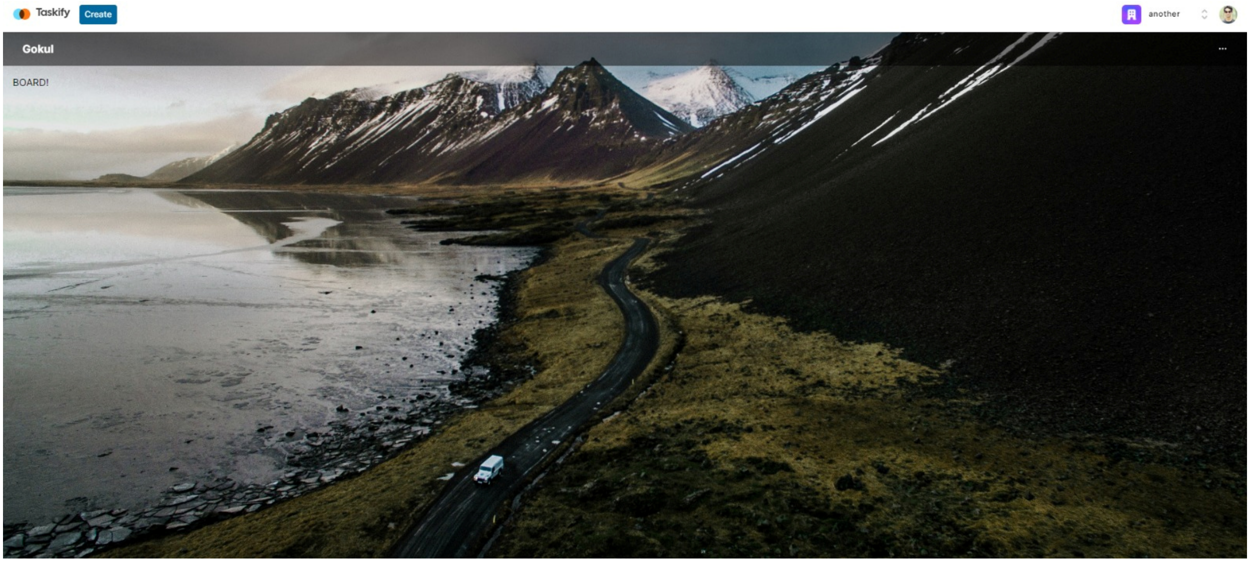


Fig. 1.5 Lists of users

ACKNOWLEDGMENT

The development of the Progress Tracker using the Next.js framework was a collaborative effort that involved the support and contributions of many individuals and organizations. We would like to express our sincere gratitude to the following:

Project Advisors and Mentors:

We are deeply grateful to our project advisors and mentors who provided invaluable guidance and expertise throughout the project. Their insights and feedback were crucial in shaping the direction and success of this project.

Development Team:

A heartfelt thank you to the development team for their dedication, hard work, and collaboration. Their technical skills and innovative thinking were essential in bringing the Progress Tracker to life.

User Participants:

We extend our appreciation to the users who participated in surveys, interviews, and testing phases. Their feedback and suggestions were instrumental in improving the application and ensuring it met user needs.

Next.js Community:

We acknowledge the support and resources provided by the Next.js community. The documentation, forums, and open-source contributions were invaluable in overcoming challenges and optimizing the application.

Family and Friends:

We are thankful to our families and friends for their unwavering support and encouragement throughout this project. Their understanding and patience were greatly appreciated during the demanding phases of development.

Educational Institutions:

We would like to thank our educational institutions for providing the necessary resources and environment conducive to research and development. The access to technical resources and academic support played a significant role in the project's success.

Technical Tools and Platforms:

Special thanks to the providers of the technical tools and platforms we utilized, including GitHub for version control, Vercel for deployment, and for database management. These tools were critical in the development and deployment processes.

Reviewers and Editors:

We appreciate the efforts of the reviewers and editors who provided constructive criticism and suggestions to enhance the quality of our work. Their attention to detail helped in refining the final output.

IV. CONCLUSION

The development and evaluation of the Progress Tracker using the Next.js framework represent a significant endeavor towards empowering individuals and teams in their pursuit of goals and milestones. Throughout this process, we have leveraged the advanced capabilities of Next.js to create a dynamic and responsive web application tailored for efficient progress tracking and goal management.

From the inception of the project, our objective has been to provide users with a seamless and intuitive experience, and the results of our evaluation affirm the effectiveness of our efforts. Through performance testing, we have ensured that the Progress Tracker delivers optimal responsiveness and scalability, capable of accommodating diverse user loads without compromising performance.

Moreover, user feedback obtained from usability testing, surveys, and interviews has been instrumental in refining the application's usability and feature set. By actively incorporating user insights into our development process, we have iteratively improved the Progress Tracker to better meet the needs and preferences of our users.

Looking ahead, the Progress Tracker stands poised for further enhancement and expansion. As we continue to gather feedback, monitor performance metrics, and iterate on improvements, we remain committed to delivering a best-in-class solution that empowers individuals and teams to visualize, track, and achieve their goals with confidence and clarity.

In conclusion, the Progress Tracker represents not only a technological achievement but also a testament to our dedication to fostering personal and professional growth. With the Next.js framework as our foundation and a commitment to continuous improvement, we are confident that the Progress Tracker will continue to serve as a valuable tool for individuals and teams striving for success in their endeavors.

V. REFERENCES

1. Next.js Documentation. (n.d.). Retrieved from https://nextjs.org/docs
2. Chart.js Documentation. (n.d.). Retrieved from https://www.chartjs.org/docs/latest/
3. Victory Documentation. (n.d.). Retrieved from https://formidable.com/open-source/victory/docs/
4. Asana. (n.d.). Task Management for Teams. Retrieved from https://asana.com/
5. Todoist. (n.d.). To Do List and Task Manager. Retrieved from https://todoist.com/
6. Trello. (n.d.). Organize Anything, Together. Retrieved from https://trello.com/
7. Slack. (n.d.). Where Work Happens. Retrieved from https://slack.com/

8. Microsoft Teams. (n.d.). Chat, Meetings, Calling, Collaboration. Retrieved from https://www.microsoft.com/en-us/microsoft-365/microsoft-teams/group-chat-software

9. Chrome DevTools Overview. (n.d.). Retrieved from https://developers.google.com/web/tools/chrome-devtools

10. Usability Testing. (n.d.). Nielsen Norman Group. Retrieved from https://www.nngroup.com/articles/usability-testing-101/

11. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "An Analytical Perspective on Various Deep Learning Techniques for Deepfake Detection", *1st International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA),* 10th & 11th June 2022, 2456-3463, Volume 7, PP. 25-30, https://doi.org/10.46335/IJIES.2022.7.8.5

12. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2022), "Revealing and Classification of Deepfakes Videos Images using a Customize Convolution Neural Network Model", *International Conference on Machine Learning and Data Engineering (ICMLDE)*, 7th & 8th September 2022, 2636-2652, Volume 218, PP. 2636-2652, https://doi.org/10.1016/j.procs.2023.01.237

13. Usha Kosarkar, Gopal Sakarkar (2023), "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", *4th International Conference on Electrical and Electronics Engineering (ICEEE)*,19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19

14. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam (2021), "Deepfakes, a threat to society", *International Journal of Scientific Research in Science and Technology (IJSRST)*, 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682

15. Usha Kosarkar, Prachi Sasankar(2021), " A study for Face Recognition using techniques PCA and KNN", Journal of Computer Engineering (IOSR-JCE), 2278-0661,PP 2-5,

16. Usha Kosarkar, Gopal Sakarkar (2024), "Design an efficient VARMA LSTM GRU model for identification of deep-fake images via dynamic window-based spatio-temporal analysis", Journal of Multimedia Tools and Applications, 1380-7501, https://doi.org/10.1007/s11042-024-19220-w

17. Usha Kosarkar, Dipali Bhende, " Employing Artificial Intelligence Techniques in Mental Health Diagnostic Expert System", International Journal of Computer Engineering (IOSR-JCE),2278-0661, PP-40-45, https://www.iosrjournals.org/iosr-jce/papers/conf.15013/Volume%202/9.%2040-45.pdf?id=7557