

My ORBII APP – A Todo Application

YAKHOOB AHMED SHARIFF¹, RUMAN PASHA², AYUB VASEEM³, NOOR AHMED BAIG⁴
^{1, 2, 3}5th Semester B.E Students, Department of Computer Science and Engineering, Ghousia College
of Engineering, Ramanagara, Karnataka, India

⁴Professor, Department of CIVIL Engineering, Ghousia College of Engineering, Ramanagara,
Karnataka, India

Abstract- MY ORBII is a modern, cross-platform task management application designed to help users organize, track, and analyze their daily tasks efficiently. Built using React Native, Expo, and TypeScript, the app provides a unified interface that works seamlessly across Android, iOS, and web platforms. It integrates Supabase as a real-time backend service with PostgreSQL as the primary database, ensuring that user data stays synchronized across devices. The application allows users to add, update, delete, and reorder tasks using an intuitive drag-and-drop interface. It also includes user authentication, cloud-based data storage, and interactive chart visualizations for task statistics. Docker and AWS are used for containerization and deployment, while GitHub Actions ensures continuous integration and testing. The project demonstrates the implementation of a scalable and responsive system that combines modern frontend frameworks, cloud databases, and DevOps tools to deliver.

I. INTRODUCTION

In today's fast-paced digital world, managing tasks effectively is crucial for productivity and organization. Mobile and web-based task management applications have become essential tools for individuals and teams to track goals and deadlines. However, many existing apps lack real-time synchronization, flexibility, or seamless cross-platform compatibility. MY ORBII addresses these challenges by leveraging modern technologies such as React Native, Expo, and Supabase to create a single app that functions across multiple platforms while maintaining a consistent user experience. It also emphasizes interactive visualization, allowing users to monitor their progress through intuitive charts and statistics. By integrating cloud computing, modern UI frameworks, and continuous deployment tools, MY ORBII exemplifies how advanced software engineering techniques can be applied to develop a robust and user-friendly productivity solution.

II. LITERATURE SURVEY & RELATED WORK

Traditional task management methods, such as manual to-do lists or standalone mobile applications, often fail to provide synchronization, scalability, and accessibility across multiple platforms. Existing systems like Google Tasks, Microsoft To-Do, and Todoist offer reliable functionality, but many of them are either platform-dependent, lack offline capabilities, or require paid subscriptions for advanced features. MY ORBII bridges these gaps by using a single, unified codebase built on React Native and Expo, ensuring consistent functionality across Android, iOS, and Web. With Supabase providing realtime synchronization and PostgreSQL ensuring reliable data management, the system overcomes the limitations of existing applications. Its integration of interactive charts, cloud authentication, and modern UI components provides users with a more visual and connected experience compared to conventional task managers.

III. SYSTEM ARCHITECTURE

MY ORBII is a cross-platform task management system designed to provide users with an efficient and intuitive way to organize their daily activities. The proposed system integrates a React Native + Expo frontend with a Supabase backend to deliver real-time synchronization and seamless performance across Android, iOS, and Web platforms. To ensure a smooth and modern user experience, React Native Paper is used for Material Design components, while React Native Chart Kit provides interactive charts for visualizing task statistics. Supabase Authentication manages secure login and registration, and Docker containerization simplifies deployment and testing. The entire system is automated through GitHub Actions for CI/CD, ensuring that every update is tested and deployed efficiently. Hosting is managed through AWS and Vercel, making MY ORBII a

reliable, scalable, and cloud-integrated productivity tool.

Software Requirements:

- Operating System: Windows / macOS / Linux
1. Frontend Framework: React Native with Expo
 2. Programming Languages: TypeScript, JavaScript
 3. Backend Services: Supabase, PostgreSQL
 4. Optional Backup Backend: Ruby on Rails
 5. Cloud & Deployment: Docker, AWS, Vercel
 6. Development Tools: Node.js, npm / Yarn, Visual Studio Code
 7. Testing Tools: Jest, React Testing Library
 8. Version Control: Git & GitHub
 9. Automation / CI-CD: GitHub Actions
 10. Infrastructure as Code: Terraform

Hardware Requirements:

1. Processor: Intel i5 or higher
2. RAM: Minimum 8 GB
3. Storage: 500 GB (minimum)
4. Mobile Devices: Android (v9.0+) / iOS (v13+) for testing
5. Internet Connectivity: Required for real-time synchronization with Supabase and AWS
6. Modules of the Project

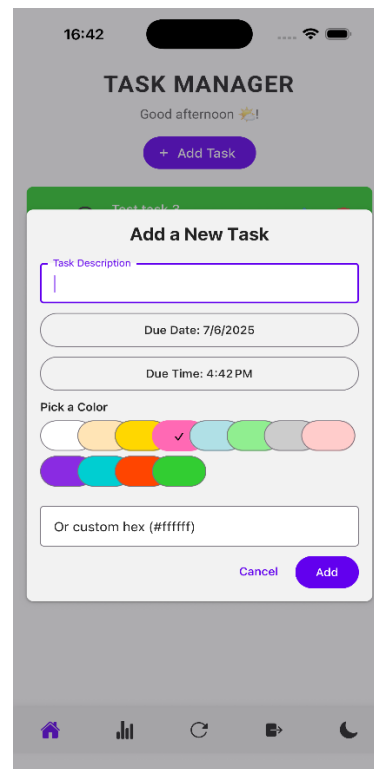
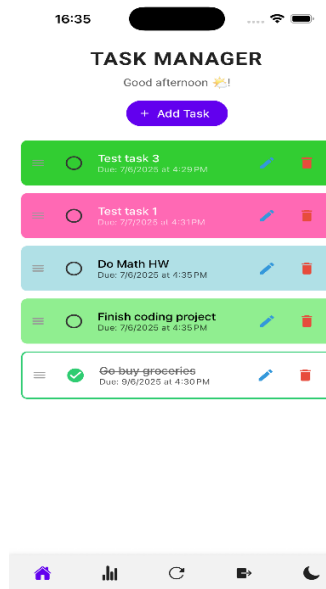
IV. METHODOLOGY

The development of MY ORBII follows a structured and modular approach, ensuring smooth integration between the frontend, backend, and deployment systems.

1. Requirement Analysis & Setup
 - o Identify user needs and define core functionalities such as task creation, updating, deletion, and synchronization.
 - o Set up the development environment using Node.js, React Native, and Expo CLI.
2. Frontend Development
 - o Design and implement the user interface using React Native Paper for Material Design and React Context API for global state management.
 - o Integrate React Native Reanimated and Gesture Handler for smooth animations and drag-and-drop task reordering.
 - o Implement theme switching (dark/light mode) for better user experience.
3. Backend Integration
 - o Configure Supabase as the backend service for authentication, database, and real-time synchronization.
 - o Use PostgreSQL for structured task and user data storage.
 - o Set up WebSocket connections to enable live updates across multiple devices.
4. Data Visualization & Analytics
 - o Implement interactive charts (Pie, Bar, and Line) using React Native Chart

Kit to display task completion and performance trends.

5. Testing & Deployment
 - o Perform unit and integration testing using Jest and React Testing Library.
 - o Containerize the app using Docker for deployment.
 - o Automate build and deployment processes through GitHub Actions, and host the web version on Vercel with backend services managed on AWS.



V. EXPERIMENTAL RESULTS

Cross-Platform Compatibility:

The MY ORBII app will run smoothly on Android, iOS, and Web platforms using a single codebase built with React Native and Expo.

- **Real-Time Synchronization:** Any updates made to tasks (add, edit, delete, or reorder) will be instantly reflected across all connected devices using Supabase Realtime.
- **User Authentication & Data Security:** Users will be able to register and log in securely, with all personal and task data stored in the cloud database (PostgreSQL) under secure authentication.
- **Interactive Data Visualization:** The app will generate dynamic charts (pie, bar, and line) to display task completion rates and productivity insights.
- **Seamless UI/UX:** Smooth navigation, responsive layout, and support for both dark and light themes will enhance the overall user experience.
- **Automated Deployment & Testing:** With Docker and GitHub Actions, the system will maintain automated builds, testing, and deployments for consistent quality control

VI. ETHICAL & PRIVACY CONSIDERATIONS

MY ORBII APP follows a responsible approach to user data, focusing on transparency, minimal data collection, and secure handling of personal information. Since it is a to-do application, users often store personal tasks, reminders, and sensitive notes, so the app ensures that all stored data is encrypted, never shared with third parties, and handled strictly for its intended functionality. The design avoids unnecessary permissions and provides clear consent mechanisms, giving users full control over what they choose to sync or store. Ethical use of technology is maintained by preventing dark-patterns, avoiding tracking-based personalization, and ensuring that users can easily delete their data at any time. Overall, the app is built to respect user autonomy, confidentiality, and digital wellbeing

VII. CONCLUSION

MY ORBII successfully demonstrates the integration of modern mobile development frameworks, cloud-based databases, and DevOps automation to build a responsive and user-friendly task management system. It provides an efficient solution for individuals and teams to manage, track, and visualize

their daily tasks in real time across multiple platforms. By combining React Native, Expo, and Supabase, the project achieves seamless synchronization and a consistent user experience on Android, iOS, and Web. The inclusion of data visualization, dark/light themes, and cloud authentication enhances both functionality and usability. Through the use of Docker, AWS, and GitHub Actions, MY ORBII also integrates modern software engineering practices such as containerization, cloud deployment, and continuous integration. Overall, this project showcases a scalable, maintainable, and visually appealing approach to cross-platform task management, setting a foundation for future enhancements like offline mode, notifications, and AI-based productivity insights

VIII. FUTURE SCOPE

Future updates for MY ORBII APP include integrating secure cloud backup with end-to-end encryption, cross-platform syncing, and optional biometric authentication for added privacy. AI-powered task recommendations and smart scheduling can be introduced while still adhering to strict ethical guidelines to prevent over-collection of data. Collaboration features—such as shared lists and team tasks—may be added with role-based access controls to maintain user privacy. Long-term, the app can expand into a broader productivity ecosystem, integrating calendars, habit tracking, and wellness insights, all while continuing to prioritize security, transparency, and user trust at every step.

REFERENCES

- [1] React Native Documentation – <https://reactnative.dev>
- [2] Expo Documentation – <https://docs.expo.dev>
- [3] Supabase Documentation – <https://supabase.io/docs>
- [4] React Native Paper – <https://callstack.github.io/react-native-paper>
- [5] AWS Cloud Services – <https://aws.amazon.com>