APP ONBOARDING FUNNEL & USER DROP-OFF ANALYSIS DASHBOARD

[1] A Rajashekar, [2] Samyuktha, [3] A Renu Sri, [4] Rohith, [5] Sudheer
[1]AssistantProfessor, [2345]Students
Department of CSE
Siddhartha Institute of Technology & Sciences, Narapally
alladirajashekar_cse@siddhartha.co.in, 23tq1a05g7@siddhartha.co.in, 23tq1a05e7@siddhartha.co.in, 23tq1a05h8@siddhartha.co.in, 23tq1a05f2@siddhartha.co.in

ABSTRACT

The App Onboarding Funnel & User Drop-off Analysis Dashboard is a data analytics and visualization system designed to understand how users interact with a mobile or web application during the onboarding process. Onboarding is the initial journey a user follows after installing or opening an application, such as registration, profile setup, permissions, feature introduction, and first successful use. Analyzing this journey helps organizations understand where users stop progressing.

The proposed system collects user-event information such as app installation, app launch, registration, login, profile completion, permission acceptance, tutorial completion, feature interaction, and successful onboarding. The system processes these events and calculates important metrics such as total users, onboarding completion rate, step-wise conversion rate, drop-off rate, average completion time, and returning-user percentage.

The dashboard represents the onboarding process as an interactive funnel. Business and product teams can observe the number of users progressing through each onboarding stage and identify stages with significant user drop-offs. Filters can be applied based on device type, operating system, application version, acquisition source, user segment, and time period.

The system also provides trend and comparison analysis to understand whether onboarding performance changes after application updates or across different user groups. Charts, graphs, KPI cards, and tables make complex user-event data easier to understand. These insights can help product teams investigate usability problems and improve the onboarding experience.

Overall, the App Onboarding Funnel & User Drop-off Analysis Dashboard provides a centralized platform for analyzing user onboarding behavior. It reduces manual analytics efforts and supports data-driven product optimization. Future enhancements can include AI-based drop-off prediction, user journey clustering, personalized onboarding recommendations, automated anomaly detection, real-time alerts, and experiment-performance analysis.

I. INTRODUCTION

Mobile and web applications compete heavily for user attention, and the first few interactions can influence whether users continue using an application. The onboarding process introduces users to the application and guides them through activities such as account creation, profile setup, permissions, tutorials, and first feature usage. A complicated or lengthy onboarding process may cause users to leave before experiencing the main value of the application.

Understanding user drop-off is therefore important for product and business teams. A user may stop onboarding because of a difficult registration process, excessive permissions, unclear instructions, technical problems, or lack of immediate value. However, identifying these patterns manually from raw application logs can be difficult when millions of events are generated.

Traditional application analytics often uses event logs, spreadsheets, database queries, and standard analytics reports. Product teams may need to create separate reports for installations, registrations, feature usage, and onboarding completion. This can make it difficult to obtain a unified view of the complete onboarding journey.

The App Onboarding Funnel & User Drop-off Analysis Dashboard provides an interactive platform for monitoring the complete onboarding funnel. It tracks users from application installation or first launch through registration, setup, tutorial completion, and first meaningful action. Users can compare funnel performance across application versions, devices, acquisition sources, and time periods.

The objective of the system is to help product teams understand where users drop off and how onboarding performance changes over time. The dashboard provides analytical evidence for further investigation and product improvement rather than assuming the exact reason for an individual user's behavior.

II. LITERATURE SURVEY

Traditional application analytics systems collect user events such as application launches, registrations, screen views, button clicks, feature interactions, and purchases. These systems provide detailed event-level information but may require additional processing to understand the complete onboarding funnel. Product teams often use analytics platforms and databases to analyze user behavior.

Funnel analysis is a widely used technique for studying sequential user journeys. In an application onboarding funnel, stages may include app installation, first launch, registration, profile completion, permission acceptance, tutorial completion, and first meaningful action. Funnel analysis helps identify the stages where users are leaving the process.

User-experience research has examined factors that influence onboarding success. Clear instructions, simple registration, limited unnecessary steps, useful personalization, fast performance, and early demonstration of product value can influence user engagement. However, the impact of these factors varies depending on the application type and target users.

Data visualization plays an important role in product analytics. Funnel charts can show conversion between onboarding stages, while line charts can display changes in completion and drop-off rates over time. Segmentation charts can compare onboarding performance across devices, application versions, acquisition channels, and user groups.

Machine learning and Artificial Intelligence have increasingly been applied to user-behavior analytics. Predictive models can estimate the likelihood of future engagement or identify patterns associated with user drop-off. AI can also support user segmentation and personalized experiences, but these approaches require appropriate validation, privacy protection, and responsible handling of user data.

Existing research and analytics platforms demonstrate the importance of event tracking, funnel analysis, and visualization for understanding user journeys. However, basic reports may provide limited cross-segment comparison and may require manual analysis to identify important onboarding changes. The proposed **App Onboarding Funnel & User Drop-off Analysis Dashboard** combines funnel monitoring, KPI analysis, segmentation, trend comparison, visualization, and reporting in a centralized system.

III. SYSTEM ANALYSIS

System Analysis

The **App Onboarding Funnel & User Drop-off Analysis Dashboard** is designed to analyze user movement through different application onboarding stages. The system

collects events such as app installation, first launch, registration, login, profile completion, permission acceptance, tutorial completion, and first meaningful action. A data-ingestion module imports event information from application analytics systems, databases, log files, spreadsheets, or supported APIs. The preprocessing module validates events, removes duplicate records, handles missing values, and standardizes timestamps and user-session information. The system organizes events according to users, sessions, onboarding stages, application versions, devices, and time periods. A KPI engine calculates total users, onboarding completion rate, stage conversion rate, drop-off rate, and average completion time. A funnel-analysis engine tracks user movement through each onboarding stage. A trend-analysis module examines changes in onboarding performance over time. A comparison engine compares aggregate drop-off patterns across devices, application versions, acquisition sources, and user segments. The visualization layer presents results using funnel charts, KPI cards, graphs, and tables. Interactive filters allow users to select specific dates, application versions, devices, and segments. Finally, product teams can generate reports and use the insights to investigate onboarding bottlenecks and improve the user experience.

Existing System

Existing applications commonly use analytics platforms, event-tracking systems, application logs, databases, spreadsheets, and manually prepared reports to monitor onboarding performance. These systems record user interactions such as app launches, registrations, screen visits, and feature usage. However, event information may be distributed across different analytics tools and databases. Product teams may need to manually define funnel stages and calculate conversion rates. Comparing onboarding performance across application versions or devices can require separate reports. Historical onboarding trends may also be difficult to analyze when event data is stored in different locations. Static reports may provide limited interactive funnel visualization. Existing systems may show individual event counts without clearly connecting them into a complete onboarding journey. Manual processing can introduce errors and consume product-team time. Reports can become outdated when new application events are generated. Identifying unusual increases in drop-off may require manual monitoring. Therefore, traditional systems provide useful event information but may require additional dashboard capabilities for comprehensive onboarding funnel analysis.

Disadvantages of Existing System

- User-event data may be distributed across multiple systems.
- Heavy dependence on manual reports.
- Time-consuming funnel calculations.
- Limited interactive onboarding visualization.
- Difficult to identify specific drop-off stages.
- Difficult to compare application versions.
- Limited device and user-segment comparison.
- Historical trends may require separate reports.
- Manual analysis can introduce errors.
- Reports may become outdated quickly.
- Limited automated drop-off alerts.
- Difficult to identify onboarding bottlenecks quickly.

Proposed System

The proposed **App Onboarding Funnel & User Drop-off Analysis Dashboard** provides a centralized platform for monitoring and analyzing user onboarding

behavior. The system collects application events from analytics platforms, databases, logs, spreadsheets, or supported APIs. A preprocessing layer validates events, removes duplicates, handles missing values, and standardizes timestamps and session information. The system organizes user activity into predefined onboarding stages such as app launch, registration, profile setup, permissions, tutorial completion, and first meaningful action. A KPI engine calculates total users, conversion rates, onboarding completion rate, drop-off rate, and average completion time. A funnel-analysis engine visually represents how users progress through each onboarding stage. A trend-analysis engine identifies changes in onboarding performance across different periods. A comparison engine allows product teams to compare aggregate onboarding performance across devices, application versions, acquisition sources, and user segments. The visualization engine presents results through funnel charts, KPI cards, line charts, bar charts, and tables. Interactive filters allow users to explore specific dates, versions, devices, and segments. The system can highlight significant changes in drop-off patterns for further investigation. Optional AI analytics can summarize aggregate trends and identify areas that may require product review.

Advantages of Proposed System

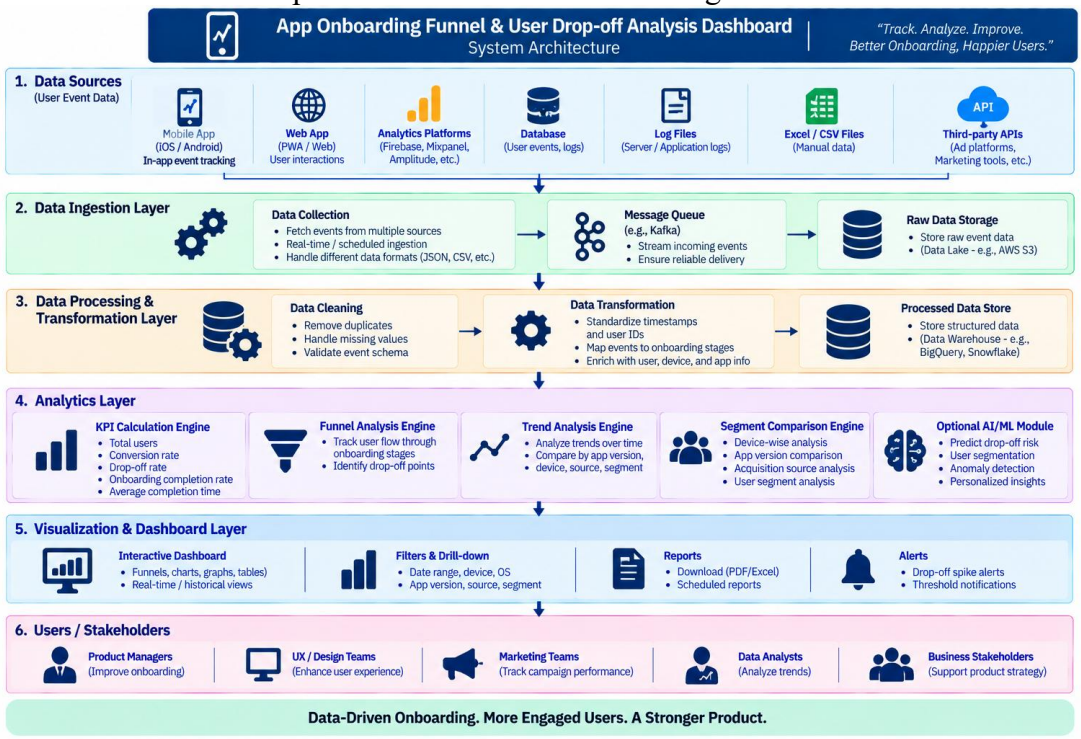
- Provides centralized onboarding analytics.
- Visualizes the complete user onboarding funnel.
- Automatically calculates conversion and drop-off rates.
- Identifies stages with high user drop-off.
- Supports application-version comparison.
- Provides device-wise analysis.
- Supports acquisition-source comparison.
- Enables user-segment analysis.
- Provides historical onboarding trends.
- Offers interactive charts and KPI cards.
- Reduces manual reporting effort.
- Supports data-driven product optimization.

IV. METHODOLOGY

The methodology begins by collecting application user-event data from analytics platforms, application databases, event logs, spreadsheets, or supported APIs. The data-ingestion module imports events such as app launches, registrations, profile completion, permissions, tutorials, and first meaningful actions. The preprocessing module validates events, removes duplicates, handles missing values, and standardizes timestamps and session identifiers. The system groups related events into user sessions and maps them to predefined onboarding stages. A KPI engine calculates total users, stage conversion rates, onboarding completion rate, drop-off rate, and average completion time. A funnel-analysis module measures the number of users progressing through each onboarding stage. A trend-analysis module examines onboarding performance across daily, weekly, monthly, and application-release periods. A comparison engine evaluates aggregate patterns across devices, operating systems, application versions, acquisition sources, and user segments. The visualization engine converts processed data into funnel charts, KPI cards, line charts, bar charts, and tables. Users can apply filters to analyze selected periods, application versions, devices, or segments. The system highlights significant changes in drop-off patterns for further investigation. Optional AI analytics can summarize aggregate trends and identify potential areas for onboarding improvement. Finally, reports are generated to support product optimization and user-experience planning.

SYSTEM ARCHITECTURE

The App Onboarding Funnel & User Drop-off Analysis Dashboard consists of several interconnected layers for collecting, processing, analyzing, and visualizing application user behavior. The User Interface allows product managers, UX teams, and analysts to access dashboards, filters, charts, and reports. The Event Data Ingestion Layer collects user events from application analytics platforms, databases, event logs, spreadsheets, and supported APIs. The Data Cleaning and Transformation Module validates events, removes duplicates, handles missing values, and standardizes timestamps and session information. The Central User Analytics Data Warehouse stores structured information about users, sessions, application versions, devices, onboarding events, and timestamps. The KPI Calculation Engine calculates user counts, conversion rates, completion rates, drop-off rates, and average onboarding time. The Funnel Analysis Engine tracks user movement from application launch through registration, setup, tutorial completion, and first meaningful action. The Trend Analysis Engine identifies changes in onboarding performance over time. The Comparison Engine compares aggregate onboarding performance across devices, versions, acquisition sources, and user segments. The Optional AI Analytics Module can summarize trends and identify unusual aggregate patterns. The Visualization Engine generates funnel diagrams, KPI cards, charts, graphs, and tables. Finally, the Onboarding Analytics Dashboard presents the analyzed information through an interactive interface for product and UX decision-making.



V. RESULT AND OUTPUT

VI. CONCLUSION

The App Onboarding Funnel & User Drop-off Analysis Dashboard provides an effective and centralized solution for monitoring user behavior throughout the application onboarding process. The system brings important information such as total installs, app launches, registrations, profile completion, permissions, tutorial completion, first key actions, and drop-off rates into a single interactive dashboard.

By using data processing, funnel analysis, KPI calculation, trend analysis, segmentation, and data visualization, the system transforms raw application event data into meaningful insights. Product managers, UX teams, and analysts can easily identify the stages where users are leaving the onboarding process and compare performance across devices, application versions, acquisition sources, and user segments.

The dashboard also helps organizations monitor changes in onboarding performance over time. Funnel charts, KPI cards, trend graphs, tables, and interactive filters make it easier to understand conversion patterns and identify potential usability bottlenecks. These insights can support improvements to registration, profile setup, permission requests, tutorials, and other onboarding activities.

Overall, the App Onboarding Funnel & User Drop-off Analysis Dashboard improves the efficiency and accessibility of user-behavior analytics. It reduces manual reporting efforts and supports data-driven decisions for improving user experience and increasing onboarding completion. Future enhancements can include AI-based drop-off prediction, personalized onboarding, automated alerts, user journey clustering, A/B testing analysis, anomaly detection, and real-time analytics.

The system should use appropriate privacy, security, access-control, and data-governance practices when handling user-event information. Analytics should be used

to identify aggregate behavioral patterns and guide product improvements rather than making unsupported assumptions about individual users.

REFERENCES

[1] Kumar, R. D., Prudhviraj, G., Vijay, K., Kumar, P. S., & Plugmann, P. (2024). Exploring COVID-19 through intensive investigation with supervised machine learning algorithm. In Handbook of Artificial Intelligence and Wearables (pp. 145-158). CRC Press.

[2] Swathi, B., Vijay, K., Sushanth Babu, M., & Dinesh Kumar, R. (2024, November). Machine Learning Techniques in Cloud Based Intrusion Detection. In The International Conference on Artificial Intelligence and Smart Environment (pp. 557-564). Cham: Springer Nature Switzerland.

[3] Sv satyakrishna, shirisha rangu ,bhargavi nalacheruve.(2024) Prospective investigation on colorectal cancer with SMOTE on machine learning Algorithm

[4] Dr.G.Vishnu Murthy, BhargaviNalacheruve 1Professor, Department of computer Science & engineering, Anurag University, TS, India. 2Student, Department of computer Science & engineering, Anurag University, TS, India.

[5] V. N. S. Manaswini, K. K, C. Nigam, S. S. Ali, R. Niranjana, and Suman, "Real-Time Object Detection in Drone Surveillance Using YOLOv5," in Proc. 2025 3rd Int. Conf. IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2025, pp. 1–6, doi: 10.1109/ICICAT68430.2025.11414670.

[6] B. Soundarya, V. N. S. Manaswini, M. Ayyakrishnan, R. D. Kumar, "Contextual Analysis of Big Data Analytics in Intelligent Transportation Frameworks," in Intersection of Artificial Intelligence, Data Science, and Cutting-Edge Technologies: From Concepts to Applications in Smart Environment, Lecture Notes in Networks and Systems, vol. 1353, Cham: Springer, 2025, doi: 10.1007/978-3-031-88304-0_79.

[7] R. D. Kumar, V. N. S.Manaswini, "Applications of blockchain in smart cities: detecting fake documents from land records using blockchain technology," in Blockchain for Smart Cities, Elsevier, 2021, pp. 105–117, doi: 10.1016/B978-0-12-824446-3.00017-X.

[8] Tejavath Veeramma, Badarla Anil, Guguloth Ravinder, "An advanced movie recommender using collaborative filtering and sentiment analysis," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 7, July 2025, doi: 10.56726/IRJMETS81618.

[9] Ravi Kumar Banoth, Ramana Murthy B V, "Automatic crop recommendation system using LightGBM and decision tree machine learning models," Journal of Machine and Computing, vol. 5, no. 1, pp. 343, Jan. 2025, doi: 10.53759/7669/jmc202505026.

[10] Ravi Kumar Banoth, Dr. B.V. Ramana Murthy, "Smart agriculture through IoT and machine learning for analyzing carbon footprints," in Proc. Int. Conf. Computer Science and Communication Engineering (ICCSCE), Apr. 2025.

[11] Ravi Kumar Banoth, B. V. Ramana Murthy, "Soil image classification using transfer learning approach: MobileNetV2 with CNN," SN Computer Science, vol. 5, art. no. 199, 2024, doi: 10.1007/s42979-023-02500-x.