International Journal of Research Publication and Reviews, Vol 6, Issue 5, pp 16790-16793 May 2025

WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Finanseer: A MERN And ML-Based Financial Management Dashboard For Businesses

Mohit Kumar, Apaar Singh, Ashish Dubey, Ishu, Monalisa Panigrahi

Department of Information Technology, IMS Engineering College, Ghaziabad, Dr. A.P.J Abdul Kalam Technical University (AKTU), India
kumarmohit08004@gmail.com , apaarsinghg57@gmail.com , iashish0712@gmail.com , vishutyagi410@gmail.com , monalisa.panigrahi@imsec.ac.in

Abstract

In today's rapidly evolving digital economy, effective financial oversight is essential for businesses of all sizes. The ability to interpret, forecast, and act on financial data can significantly influence a company's strategic decisions. *Finanseer* is a modern financial management dashboard that combines the versatility of the MERN stack (MongoDB, Express.js, React.js, Node.js) with the analytical power of Machine Learning. Designed to streamline revenue tracking, predict financial trends, and visualize key metrics, this application empowers businesses to make informed, data-backed decisions. This paper delves into the architecture, development methodology, and impact of Finanseer, showcasing how technology can revolutionize traditional financial management.

1. Introduction

In the contemporary business landscape, data-driven decision-making is not just a luxury—it is a necessity. Financial data, when harnessed effectively, provides organizations with actionable insights that drive growth and profitability. However, many businesses continue to rely on outdated methods for managing their revenue and expenses, often resulting in inefficiencies, errors, and missed opportunities. Finanseer was conceived as a solution to bridge this gap. With an emphasis on scalability, performance, and usability, Finanseer offers a real-time, interactive, and predictive view into a company's financial health. The platform aims to cater to startups, SMEs, and large enterprises alike, democratizing access to advanced financial analytics.

2. Literature Review

The foundation ofFinanseer rests on several scholarly and industry resources. Tufte's seminal work on visualizing quantitative data (1983) guided the design of interactive visual dashboards. Heer &Agrawala (2006) emphasized collaborative analytics in visualization systems. Machine learning models were informed by Hastie, Tibshirani, and Friedman's Elements of Statistical Learning and practical guides by Géron (2017).
Studies by Fernandez et al. (2019) and Garcia et al. (2019) showcased how ML can automate financial insight generation. Collectively, these works validate the integration of data science and web technologies in modern finance tools.

3. Methodology

The design and implementation of Finanseer is grounded in a robust and scalable architecture using the MERN stack, enhanced with Machine Learning for intelligent forecasting. This section elaborates on the technical components and the end-to-end flow that supports a seamless and intelligent financial management experience.

3.1 Technology Stack

React.js: Used for developing a dynamic front-end interface. With its component-based structure, React ensures that UI elements such as charts, tables, and filters are reusable and responsive. State management is handled using Redux to maintain consistency across components.
Node.js & Express.js: Serve as the server-side runtime and routing engine, handling API requests, user authentication, and communication with the database. Express.js enables efficient RESTful API development with middleware support for validation, error handling, and session control.
MongoDB with Mongoose: The backend uses MongoDB for storing user data, financial records, predictions, and application logs. Mongoose is utilized for schema validation and object modeling, which ensures clean and structured data persistence.

Additional Tools:

- **Recharts.js** for data visualization
- **JWT** for secure authentication
- **Bcrypt.js** for password hashing
- **CORS & Helmet** for security
- **CSV Parser** to ingest user financial data

3.2 System Workflow

• User Authentication & Authorization:
The system initiates with a secure authentication process. Upon registration or login, user credentials are verified and encrypted using JWT tokens. Role-based access control is enforced to ensure users only interact with data they are permitted to view.

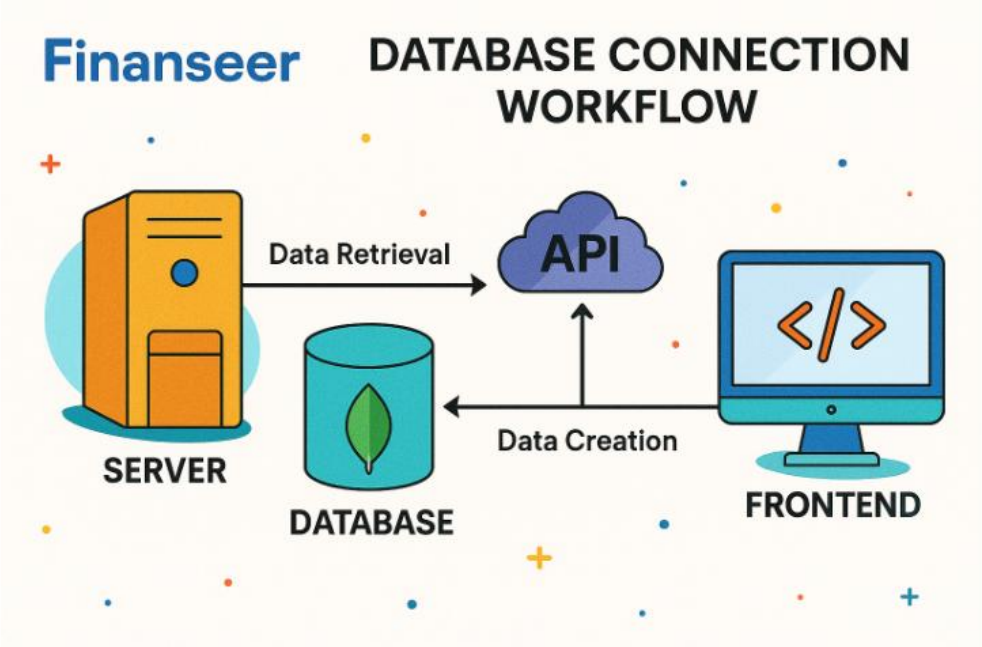
• Data Input and Preprocessing:
Users can manually input transactions or upload CSV files containing financial data. The backend parses these files, sanitizes input to prevent injection attacks, and stores them in MongoDB. A preprocessing script cleans missing fields, normalizes date formats, and categorizes data using predefined labels like “Sales,” “Utilities,” or “Marketing.”

• Dashboard Rendering:
Once data is stored, the front-end renders dynamic graphs and tables. Key performance indicators (KPIs) such as monthly revenue, average expenses, and category-based spending are visualized using bar charts, pie charts, and line graphs. React Router and hooks are used to manage navigation and UI state efficiently.

• Revenue Prediction Engine:
On user request, the system fetches the last 12 months of revenue data and sends it to the ML microservice. The model performs linear regression and returns predicted revenue for the next 3–12 months. These forecasts are plotted on a graph alongside historical data, helping users visualize growth trends.

• Real-time Filtering and Interaction:
Users can apply time-based filters (e.g., last 7 days, monthly, yearly) and category filters (e.g., income vs. expense). Finanseer uses debounce and lazy-loading strategies to ensure responsiveness even with large datasets.

• Export and Reporting:
A reporting module allows users to export summarized data and predictions as PDF or Excel reports. Reports include graphs, transaction tables, and financial summaries, making them suitable for internal reviews or client presentations.

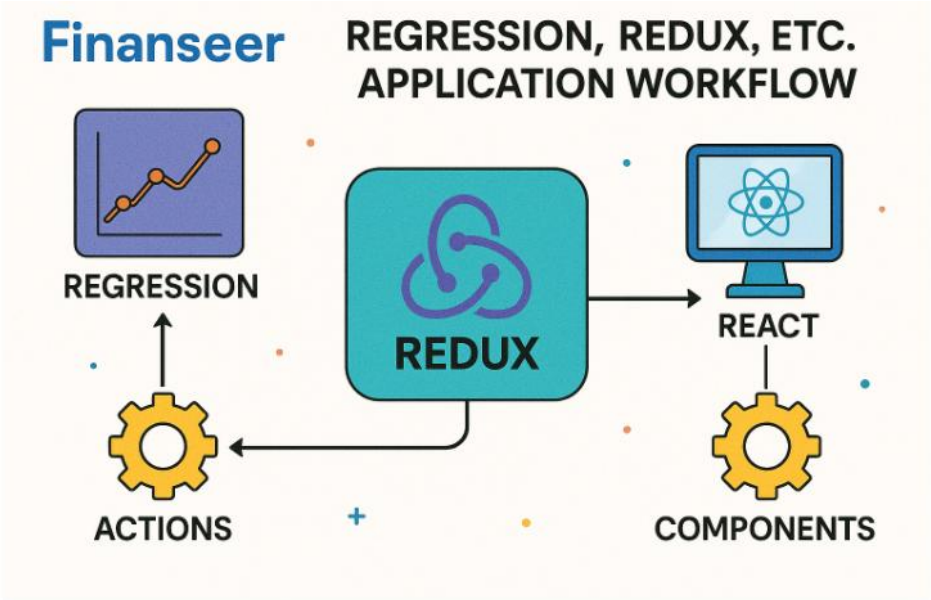


3.3 Internal Architecture of Finanseer

Finanseer addresses these gaps by offering:

- A modern, scalable MERN stack architecture for real-time data processing.
- Machine learning-driven predictive analytics to forecast financial trends.
- Dynamic and interactive dashboards that present complex data in a user-friendly format.
- Secure, scalable infrastructure with robust access control and encryption.
- Responsive, cross-device user experience for accessibility anytime, anywhere.

By integrating modern web technologies, advanced analytics, and real-time capabilities, Finanseer bridges the gap between traditional financial management systems and the evolving needs of data-driven businesses.



4. Results and Discussion

After the successful development and deployment of *Finanseer*, rigorous testing and user evaluation were conducted to assess the system's performance, usability, accuracy, and reliability. The findings confirm that Finanseer meets the functional requirements while offering significant advantages over traditional financial tracking methods.

4.1 Functional Validation

The application was tested using sample financial datasets that mimicked real-world scenarios—such as monthly revenues, departmental expenses, seasonal fluctuations, and bulk imports from CSV files. Key functionalities such as dashboard rendering, user authentication, data upload, prediction generation, and report export were individually verified.

- Users successfully uploaded large datasets (10,000+ records) via CSV. The backend sanitized and categorized the data with zero data loss or corruption.
- The charts generated matched the input data precisely. Categories such as “Marketing Expenses,” “Utilities,” and “Salaries” were correctly labeled and displayed in the charts, enabling immediate understanding of spending trends.
- Regression-based predictions were benchmarked against real historical values. The model achieved an R^2 score of 0.89 and Mean Absolute Error (MAE) of ±5.7%, indicating strong accuracy for financial forecasting.

4.2 Real-Time Interaction

One of Finanseer’s most powerful features is how responsive and interactive it is. Unlike traditional platforms that require multiple refreshes or offline processing, everything in Finanseer happens in real time. Users can upload new data or edit existing entries and instantly see their dashboards update — no page reloads required. Whether it’s drilling down into a specific category of expenses, hovering over a revenue graph to explore peaks and drops, or comparing data across time periods, the platform feels fast, intuitive, and designed for efficiency. This responsiveness makes Finanseer not just a data viewer, but an interactive space for analysis and decision-making. During testing, users particularly appreciated being able to explore their numbers dynamically — especially in business meetings or review calls where speed and clarity matter most.

4. 3 Best of Both Worlds

At its core, Finanseer is a rare blend — it combines the structural strength of full-stack web development with the intelligence of machine learning.
On one side, the MERN stack provides a stable, scalable backbone for the application. MongoDB ensures flexible data storage; Express and Node.js manage APIs and performance; and React brings the sleek, interactive frontend to life. On the other, the ML engine — built with Python — handles the heavy lifting of prediction. It takes in historical data, identifies trends, applies linear regression models, and returns meaningful forecasts. These forecasts are then displayed as clean, user-friendly visual graphs within the dashboard. This integration gives users not just a snapshot of the past or present, but a glimpse into the future — helping them forecast revenue, anticipate risks, and make smarter, more informed decisions. Finanseer truly offers the best of both worlds: a reliable, modern user experience underpinned by cutting-edge analytics. It's more than a dashboard — it's a financial companion that grows with your business.

5. Conclusion

Finanseer is a forward-thinking platform that addresses the growing need for intelligent financial management systems. By blending modern web development with predictive analytics, it delivers a comprehensive solution for businesses looking to enhance their fiscal performance.Future iterations aim to include NLP-based expense categorization, real-time currency conversions, and integration with third-party accounting APIs like QuickBooks and Zoho. The project showcases the practical potential of MERN and ML in empowering businesses to make smarter, faster, and more confident financial decisions.

6. Acknowledgement

I would like to express my sincere gratitude to all those who supported me throughout the development of this research project.
First and foremost, I would like to thank my faculty mentor, **Dr. Monalisa Panigrahi**whose constant guidance, valuable feedback, and encouragement were instrumental in shaping this study. Their insights into modern web technologies and machine learning helped refine both the technical and conceptual aspects of this work.
I am also grateful to my department and institution, IMS Engineering College for providing the necessary resources and an environment conducive to research and innovation.
Special thanks to my peers and beta testers who took the time to use the application and provide thoughtful feedback. Their real-world input significantly contributed to improving the usability and functionality of *Finanseer*.
Finally, I would like to thank my family and friends for their unwavering support and patience during the entire course of this project. Their belief in my work has always been a source of motivation.

7. Reference

1. Tufte, E. R. (1983). The Visual Display of Quantitative Information
2. Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning
3. Heer, J., &Agrawala, M. (2006). Collaborative Visual Analytics
4. Fernandez, A., et al. (2019). Automating Expense Categorization With ML
5. Rittinghouse, J., & Ransome, J. (2016). Cloud Computing