

Final Project Report

Project: Cost Wise Innovators – Cost Optimization Dashboard

Team: CSDD1

Date: April 1, 2025

Submitted to : Michael Del Rosario

1. Executive Summary

The Cost Wise Innovators project aimed to design and develop a scalable, interactive cost optimization dashboard to help users monitor, analyze, and control their expenses effectively.

The core idea was to offer a modern, mobile-friendly, full-stack web application that simplifies expense tracking and enhances financial decision-making.

Over the course of the project, we gathered requirements, designed UI/UX wireframes, built the application using modern technologies, conducted testing and validation, and finally deployed a working product.

The outcome is a user-centric solution that combines data visualization, analytics, budget alerts, and reporting tools in one seamless platform.

2. Project Objectives

The major objectives of this project were as follows:

- Build an end-to-end expense monitoring system
 - Ensure the application is intuitive and accessible on all devices
 - Implement monthly and category-based budget features
 - Allow users to visualize their financial patterns through charts
 - Ensure user data is protected through secure login and encrypted storage
 - Enable export of insights in common file formats (PDF, Excel) for offline usage
 - Provide real-time feedback and alerts for overspending scenarios
-

3. Project Scope

This dashboard was designed for individuals, freelancers, and small businesses to manage finances in real-time.

The scope included the implementation of frontend and backend systems, secure user authentication, visualizations (pie charts, bar graphs, line charts), a monthly budget planner, a report generator, and admin-level configurations.

The scope did not include real-time banking APIs, multiple language support, tax calculations, or AI-driven financial forecasting, although these features are planned for future iterations.

4. Technologies Used

- **Frontend:** React.js for building the user interface, with responsive design powered by CSS3 and modular components
 - **Backend:** Node.js and Express.js were used for RESTful APIs and server-side logic
 - **Database:** Firebase Realtime Database and MongoDB Atlas were tested for performance and scalability
 - **Authentication:** Firebase Authentication ensured secure, password-protected user sessions
 - **Visualization:** Chart.js was used to render pie charts, bar graphs, and line graphs for expense tracking
 - **Hosting:** Vercel and Firebase Hosting were used for frontend deployment, ensuring continuous integration
 - **Design & Prototyping:** Figma was used to develop UI/UX wireframes prior to coding
 - **Version Control:** Git and GitHub supported collaborative development
-

5. System Design

The architecture followed a decoupled frontend-backend model:

- Frontend communicates with backend APIs to retrieve and update data
- Backend APIs handle user validation, CRUD operations on expense records, and budget calculations
- All data is persisted in a cloud database (Firebase or MongoDB)
- JWT tokens ensure session management and secure API communication

The design adhered to principles of modularity, scalability, and reusability. Each module (dashboard, expenses, budget, report) is self-contained and testable.

6. Implementation Highlights

- A single-page application with dynamic state updates (React Context API)
 - Fully responsive layout optimized for phones, tablets, and desktops
 - Budget alerts with threshold markers (80% = yellow, 100% = red)
 - Role-based access layer for future scalability (admin vs. user)
 - Export options with downloadable reports in PDF and Excel formats
 - Integration of QR code linking for real-time mobile access
-

7. Testing and Validation

Testing was conducted at multiple levels:

- **Unit Testing:** Each component (AddExpense, BudgetTracker, Charts) was independently tested
 - **Integration Testing:** End-to-end flow tested from login to report generation
 - **Cross-Browser Testing:** Validated compatibility across Chrome, Firefox, Edge, and Safari
 - **Mobile Responsiveness:** Ensured UI scaled appropriately on Android and iOS devices
 - **Security Testing:** Validated Firebase rules, route protection, and password encryption
-

8. Challenges Faced

- Rendering real-time updates in charts without reloading the page
 - Synchronizing data between state and database in asynchronous flows
 - Designing an intuitive UI that remains functional across screen sizes
 - Merging code conflicts during Git-based collaboration
 - Managing time across sprints and distributing tasks evenly among members
-

9. Final Outcomes

- Successfully deployed a cost optimization dashboard
 - Achieved all milestones and presented a working demo at EXPO
 - User interface received positive feedback for simplicity and design
 - Application can serve as a foundation for future financial products
 - Team demonstrated capability in full-stack development, agile delivery, and effective teamwork
-

10. Lessons Learned

- Importance of documenting user stories and test cases early
 - Agile boards (Trello) helped track sprint progress and task dependencies
 - Real-time collaboration tools (Google Meet, GitHub) reduced delays
 - UI feedback loops led to rapid design improvements
 - Regular code reviews and testing ensured higher quality delivery
-

11. Future Enhancements

- Integration with third-party banks to automate transaction imports
 - AI-generated savings recommendations based on user patterns
 - Multi-user collaboration for household or team budgets
 - Gamified goals and rewards for staying under budget
 - Native mobile app (React Native) to increase offline accessibility
-

12. Conclusion

The Cost Wise Innovators Dashboard is a solid example of applying modern software practices to solve real-world financial problems.

Our group successfully delivered a practical, scalable, and visually appealing product that demonstrates the power of collaboration, agile development, and thoughtful design.

We are proud of our results and look forward to evolving this solution further.