



RESEARCH ARTICLE

Development of Budget Management System Using Visual Basic .NET and MySQL Database: A Desktop Application for Personal Financial Tracking

Janobe ^{1*}

¹ Independent Developer, SourceCodester Community, Philippines.

Correspondence

^{1*} Independent Developer, SourceCodester Community, Philippines.
Email: janobe@budgetmanagement.com

Funding information

Independent Developer, SourceCodester Community, Philippines.

Abstract

This study developed a desktop-based Budget Management System using Visual Basic .NET 2015 and a MySQL database to address the growing need for simple, secure, and offline financial management tools. The system enables users to record, update, and review income and expenses while automatically generating summaries of total income, total expenses, and current balance. Designed under the Software Development Life Cycle (SDLC) framework, it prioritizes usability, reliability, and privacy by storing all data locally and eliminating dependency on internet connectivity or subscription-based platforms. Testing results confirm that the system performs efficiently, remains stable even with large datasets, and is intuitive for users of varying technical backgrounds. Beyond its practical value, the system serves educational and developmental purposes by illustrating principles of database integration, user interface design, and software engineering in Visual Basic .NET. Although current limitations include manual installation and single-user operation, these provide clear directions for future improvement, such as automated setup, cross-platform compatibility, and advanced visualization features. Overall, the project demonstrates that accessible and privacy-oriented technology can significantly improve personal financial management while promoting financial literacy and independence.

Keywords

Budget Management System; Visual Basic .NET; MySQL; Financial Literacy; Desktop Application.

1 | INTRODUCTION

Effective financial management has become an essential skill in an era characterized by economic instability, rising living costs, and employment uncertainty. The increasing financial pressures faced by households globally have underscored the importance of systematic approaches to budgeting and expenditure control. Studies indicate that individuals who actively manage their finances are more likely to achieve financial stability and make sound economic decisions (Lusardi & Mitchell, 2014; Fernandes, Lynch, & Netemeyer, 2014; Xiao & O'Neill, 2016). However, many still struggle to maintain consistent financial discipline, often due to limited access to reliable tools for recording, monitoring,

and analyzing personal financial transactions (Ameriks, Caplin, & Leahy, 2003; Shim *et al.*, 2010). Technological advancement has made financial management more accessible through the development of budget management systems and financial tracking software. Early research in this domain has highlighted the potential of combining programming languages with relational databases to create user-oriented systems that enhance the accuracy and efficiency of financial record keeping (Patel & Patel, 2016). The integration of programming environments such as Visual Basic .NET with databases like MySQL enables the creation of applications that automate data handling, reduce manual computation, and provide real-time feedback to users (Oracle Corporation, 2023; Microsoft Corporation, 2015). Prior works, including those by Andriana *et al.* (2021) and Tafonao *et al.* (2023), demonstrated the utility of database-driven systems for managing organizational budgets and local government finances, emphasizing structured data handling and accuracy in reporting. Similarly, studies by Riadi *et al.* (2024), Setiawan (2025), and Ultariani *et al.* (2020) have illustrated how Visual Basic and MySQL-based applications can efficiently support financial and operational processes within various sectors. Although these projects focused primarily on institutional or business contexts, their methodological foundations can be adapted for personal financial management. The development of a desktop-based budget management system offers several advantages over cloud-based alternatives. Unlike web or mobile applications that require continuous connectivity and often involve third-party data storage, desktop systems maintain user privacy by storing financial records locally. This model is particularly relevant in regions where internet access remains limited or unreliable. Visual Basic .NET provides a versatile environment for designing user-friendly interfaces and implementing data operations through the ADO.NET framework, while MySQL ensures structured data management and fast query execution (Troelsen & Japikse, 2017; Oracle Corporation, 2023). The combination of these technologies supports the creation of applications that are functional, secure, and adaptable for personal use.

Traditional methods such as handwritten ledgers or basic spreadsheets often fail to meet the expectations of modern users who seek immediacy, automation, and accuracy in financial analysis. These manual approaches are time-consuming and prone to error, limiting their effectiveness for users who require dynamic and reliable insights into their income and expenses (Grable & Joo, 2004). Contemporary research in behavioral economics also suggests that accessible budgeting tools can encourage more deliberate spending and saving behavior by enhancing individuals' awareness of their financial status (Thaler & Sunstein, 2008; Ramsey, 2013). The current study builds on this understanding by developing a desktop-based Budget Management System using Visual Basic .NET 2015 and MySQL, designed to help users systematically record income and expenses, track financial trends, and manage their budgets with improved precision. The system is structured to operate offline, providing a private and cost-free alternative to commercial financial management software. It enables users to perform core operations such as recording, updating, deleting, and reviewing financial transactions through a simple and organized interface. Moreover, the system generates summaries of total income, total expenses, and current balance, allowing users to evaluate their financial position at any time. Its development follows the Software Development Life Cycle (SDLC) methodology, ensuring systematic planning, implementation, and testing throughout all phases. From a broader perspective, the system not only addresses the practical need for accessible financial tools but also contributes educational and developmental value. It serves as an applied model for students and novice programmers learning how to integrate Visual Basic .NET with MySQL for desktop applications. Similar to the work of Andriana *et al.* (2021) and Tafonao *et al.* (2023), this research emphasizes the balance between technical reliability and usability, aiming to produce a resource that is both functionally effective and pedagogically instructive. The open-source nature of the system further encourages collaborative enhancement and adaptation across different contexts, fostering innovation in financial technology development. Ultimately, the evolution of budget management systems reflects a continuing shift toward technological empowerment in personal finance. By combining intuitive programming environments with robust database management systems, such tools enable individuals to manage resources more strategically and strengthen their financial independence (Patel & Patel, 2016; Lusardi & Mitchell, 2014). The present research therefore seeks to advance this paradigm through the development of a Budget Management System that is efficient, private, and adaptable to diverse user needs.

2 | LITERATURE REVIEW

The foundation of this research lies in previous studies that explore budget management principles, personal finance systems, and the technological frameworks that support them. Budget management is broadly defined as a structured process of planning, organizing, monitoring, and controlling financial resources to achieve financial stability and long-term goals. Scholars in financial literacy and behavioral economics emphasize that individuals who consistently record and evaluate their income and expenses tend to make more rational financial decisions, maintain savings habits, and reduce exposure to debt (Lusardi & Mitchell, 2014; Fernandes, Lynch, & Netemeyer, 2014; Xiao & O'Neill, 2016). Earlier methods such as manual record-keeping or spreadsheet-based tracking, though useful, have proven insufficient in providing the accuracy, immediacy, and automation required in contemporary financial management. Over time, these practices have evolved into more systematic and technology-assisted approaches that incorporate behavioral analysis and predictive

functions, reinforcing what researchers describe as the *tracking effect*—a behavioral outcome in which consistent monitoring enhances awareness and financial discipline (Ameriks, Caplin, & Leahy, 2003; Shim *et al.*, 2010). Technological innovation has significantly shaped the evolution of personal finance management systems. Early software relied on manual data input and user-generated formulas, offering marginal improvements over traditional methods. The introduction of specialized software in the late twentieth century, such as *Quicken* and *Microsoft Money*, introduced automated transaction categorization and integration with banking institutions, marking a turning point in financial technology. However, these systems often entailed complex interfaces and high costs that limited their accessibility for average users. The subsequent proliferation of mobile technology and cloud computing brought about a new generation of applications such as *Mint*, *YNAB (You Need A Budget)*, and *PocketGuard*, which provide real-time transaction synchronization, goal tracking, and machine learning-based expense categorization. Although such applications enhance convenience and analytical capability, they also raise persistent concerns about data privacy, dependency on third-party services, and the long-term affordability of subscription-based business models (Thaler & Sunstein, 2008; Ramsey, 2013). Consequently, there remains a clear need for offline, user-controlled systems that protect financial data while remaining functionally reliable and easy to use.

The technological frameworks most relevant to this research—Visual Basic .NET and MySQL—offer practical and sustainable solutions for building such systems. Visual Basic .NET, an object-oriented programming language within Microsoft's .NET Framework, facilitates the rapid development of desktop applications with intuitive graphical interfaces (Microsoft Corporation, 2015; Troelsen & Japikse, 2017). It provides robust libraries for user interface design, data connectivity through ADO.NET, and comprehensive error handling, making it an ideal platform for developers seeking efficiency and reliability. MySQL, developed and maintained by Oracle Corporation, complements Visual Basic .NET as an open-source relational database management system recognized for its scalability, reliability, and support for standard SQL operations (Oracle Corporation, 2023). The integration of Visual Basic .NET with MySQL, typically implemented through the MySQL Connector/.NET driver, allows seamless communication between front-end and back-end components. This combination is widely adopted in academic and commercial projects due to its flexibility, community support, and suitability for small to medium-sized applications (Patel & Patel, 2016). Prior research in this area has also highlighted the importance of system usability and user-centered design. Studies by Andriana *et al.* (2021) and Tafonao *et al.* (2023) demonstrate that systems built with Visual Basic .NET and MySQL can enhance budget accuracy and transparency within institutional or governmental settings. Similarly, Riadi *et al.* (2024), Setiawan (2025), and Ultariani *et al.* (2020) have shown that the same technological combination effectively supports transactional and operational applications in business and retail contexts. These findings affirm the technical robustness of Visual Basic .NET and MySQL integration while revealing a gap in applications tailored specifically to personal finance. User experience research further suggests that interface simplicity plays a crucial role in determining long-term adoption and effectiveness. Applications that minimize the cognitive and procedural “friction” involved in data entry are associated with higher engagement and improved financial outcomes. In this respect, the development of intuitive, focused tools that streamline interaction without unnecessary complexity remains a vital direction in financial software design.

Desktop-based systems continue to hold unique advantages despite the rise of web and mobile alternatives. They can function offline, provide faster local processing for data-intensive tasks, and preserve complete user control over stored information. These attributes are especially valuable for users in regions with unreliable internet connectivity or heightened privacy concerns. Furthermore, desktop applications avoid recurring subscription costs, making them more inclusive for users with limited financial means. However, while many studies have validated the technical and behavioral benefits of financial management systems, relatively few have explored solutions that merge usability, offline operation, and privacy preservation within a single, open-access platform. The current research responds to this gap by developing a desktop-based Budget Management System that integrates the dependability of Visual Basic .NET and MySQL with an emphasis on simplicity and user autonomy. In summary, the literature demonstrates a clear progression from manual financial tracking methods to sophisticated, technology-driven systems that leverage software integration to improve decision-making and financial literacy. Prior research provides a strong empirical and technological foundation for developing practical financial management tools, *yet also* highlights the limitations of existing commercial and cloud-based solutions. By synthesizing these insights, the present study builds on established approaches to deliver an offline, accessible, and privacy-conscious budget management system designed to empower individuals to monitor and control their finances effectively.

3 | METHODOLOGY

The development of the Budget Management System was guided by a systematic approach rooted in established software engineering principles to ensure functionality, reliability, and user-centered design. This section outlines the research design, system architecture, database structure, implementation strategy, testing procedures, and the tools and technologies utilized throughout the development lifecycle. The chosen methodology aimed to produce a robust

application capable of meeting user needs while maintaining technical integrity and usability. This study employed a developmental research design, an approach particularly suited for creating and evaluating new technological solutions. It emphasizes iterative refinement through continuous design, development, testing, and evaluation, with each phase contributing insights that inform subsequent improvements. The project followed the Software Development Life Cycle (SDLC) framework, which provides a structured process consisting of six key stages: planning, analysis, design, implementation, testing, and deployment. Each phase contributed to ensuring that the system evolved logically from conceptualization to completion. The planning phase focused on identifying core requirements based on user expectations and a review of existing budget management systems. This stage established the system's main objectives, which included enabling users to record and track income and expenses, view transaction histories, modify data entries, and generate financial summaries efficiently. Following this, the analysis phase examined both functional and non-functional requirements. The functional requirements defined the core system behaviors such as recording transactions with relevant details—amounts, remarks, transaction types, and dates—along with the ability to update, delete, filter, and calculate income and expenditure totals.

Non-functional requirements addressed performance expectations including interface usability, data integrity, reliability, and maintainability to ensure smooth operation and scalability for future enhancement. The design phase translated these specifications into an integrated system structure encompassing both architecture and interface layout. The application adopted a two-tier client-server architecture, separating the presentation and data layers to achieve a balance between simplicity and efficiency. The presentation layer, implemented using Visual Basic .NET, managed user interaction through forms, controls, and event-driven operations, while the data layer, built on MySQL, handled data storage, retrieval, and integrity management. This separation ensured modularity, making the system easier to maintain and extend. Database design was a critical component of the development process. The database, titled *db_budget*, was designed to support transaction management efficiently while maintaining accuracy and scalability. Its core table, *transactions*, included several key fields: *transaction_id* as a primary key with auto-increment functionality for unique identification; *amount* as a decimal field for precise financial values; *remarks* as a variable character field allowing users to add context to transactions; *transaction_type* as an enumerated field limited to "Received" and "Expense" for consistency; and *transaction_date* for chronological organization and filtering. An additional *created_at* timestamp automatically recorded entry times to provide an audit trail, thereby improving transparency and traceability. The implementation phase transformed the design into a fully functional system coded in Visual Basic .NET 2015 using the Microsoft Visual Studio IDE. The MySQL database was accessed via ADO.NET and the MySQL Connector/.NET driver, ensuring secure and efficient data communication. The system interface was divided into intuitive sections: an input area for entering transaction data (including textboxes for amount and remarks, a combobox for transaction type, and a date picker); an action section with buttons for saving, updating, deleting, clearing, and searching records; a data display section that utilized a DataGridView control to present stored transactions in tabular form; and a summary panel that dynamically calculated total income, total expenses, and the current balance. Core functionalities were implemented to optimize user experience and data integrity.

The Save feature validated input data before executing parameterized SQL *INSERT* commands to prevent SQL injection, while the Update feature allowed modifications to existing transactions through SQL *UPDATE* queries. The Delete feature included confirmation prompts to prevent accidental data loss, and the Search function allowed users to filter transactions within specific date ranges using *WHERE* clauses. Summary calculations employed SQL aggregate functions to compute totals and current balances in real time, while color-coded visual cues—green for surplus and red for deficit—provided immediate feedback on financial status. The testing phase verified that the system met its design objectives and functional requirements through a series of rigorous evaluations. Unit testing was used to confirm that individual modules performed as expected. Integration testing assessed the coordination between the Visual Basic .NET interface and the MySQL database, ensuring seamless communication and data consistency. Functional testing confirmed that features such as transaction recording, modification, and filtering operated correctly. Usability testing, conducted with a sample group of users, evaluated the clarity of the interface and revealed opportunities for improvement in navigation and visual presentation. Performance testing assessed system responsiveness under varying loads, confirming that the system maintained efficient processing speeds even when managing large datasets. The selection of development tools and technologies was based on practicality, stability, and accessibility. Microsoft Visual Studio 2015 was used as the integrated development environment for its comprehensive debugging and project management capabilities. Visual Basic .NET served as the primary programming language, chosen for its simplicity and strong graphical interface support. MySQL 5.7 or higher was selected for reliable database management, while XAMPP simplified database configuration and hosting through its bundled Apache and MySQL services. phpMyAdmin was utilized as an auxiliary tool for managing and testing database operations during both development and deployment phases. System requirements were defined to ensure accessibility across a broad range of personal computers.

Hardware requirements included a processor equivalent to Intel Core i3, at least 4GB of RAM, 500MB of available storage, and a display resolution of 1024x768 pixels. The software prerequisites included Windows 7 or higher, .NET Framework 4.5 or newer, and an installed MySQL environment provided through XAMPP or similar packages. Deployment

was designed to be user-friendly yet technically sound. The system was distributed as a compressed archive containing the Visual Basic .NET project files, executable program, SQL script for initializing the database, and documentation with installation and usage instructions. To activate the application, users were required to install XAMPP, create a database named *db_budget* using phpMyAdmin, import the SQL script, and run the executable. Once connected to the database, the system operated independently of internet connectivity, providing a fully offline solution. Throughout development, strong emphasis was placed on code quality, maintainability, and error resilience. Modular programming principles were applied, ensuring that each function served a distinct purpose. Meaningful variable names and inline documentation were included to enhance readability and facilitate future updates. Error-handling mechanisms were implemented across all major operations to prevent unexpected crashes and to provide informative feedback to users. In conclusion, the methodology combined systematic analysis with iterative development to create a reliable and efficient Budget Management System. The use of the SDLC framework ensured that each phase—from planning to deployment—was executed in an organized and traceable manner. The integration of Visual Basic .NET and MySQL provided a strong technical foundation, while the testing and design principles ensured that the final application achieved both functionality and user accessibility. This methodological rigor establishes a solid base for future enhancements as user requirements evolve and new technologies emerge.

4 | RESULTS AND DISCUSSION

4.1 Discussion

The implementation of the Budget Management System produced a fully functional desktop application that meets the predefined objectives and functional requirements. The completed system provides users with an intuitive and visually organized interface, ensuring that all essential operations—such as data entry, modification, and analysis—can be performed efficiently within a single dashboard. This section outlines the design features, functional components, and performance outcomes resulting from the development and testing stages. Upon launching the application, users are presented with a main dashboard that consolidates all financial management tasks in one workspace. The input panel, located at the top of the interface, contains clearly labeled fields for entering transaction details. Users can input the transaction amount through a numeric textbox with built-in validation to prevent invalid entries. Descriptive notes can be added in the remarks field, which helps users recall the nature of each transaction, such as “Salary,” “Groceries,” or “Electricity bill.” The transaction type dropdown menu offers two choices—Received for income and Expense for expenditure—ensuring consistent categorization. The integrated date picker automatically selects the current date by default but also allows modification for retroactive or future entries, enabling flexible data management. Directly below the input area lies the action toolbar, which includes five core buttons: Save, Update, Delete, Clear, and Search.

Each button corresponds to a key function in the system. The Save button validates all inputs before executing a parameterized SQL INSERT command that adds the new record to the MySQL database, preventing SQL injection vulnerabilities. After a successful entry, the form clears automatically, and the transaction list updates in real time. The Update button activates when an existing transaction is selected from the table, allowing users to revise details without altering the transaction ID. The Delete button removes a transaction after a confirmation prompt, reducing accidental deletions. The Clear button resets all fields to their default state, while the Search button triggers date-based filtering, allowing users to specify start and end dates for transaction analysis. The central component of the interface is the Data Display Table, which uses a DataGridView control to present all recorded transactions in an organized tabular format. Each row represents a financial entry, with columns for transaction ID, amount, type, date, and remarks. Transactions are displayed with clear formatting, including currency symbols and date alignment for readability. Users can sort the table by any column and select specific rows to modify or delete entries. The table updates automatically after each operation, ensuring data consistency without manual refresh.

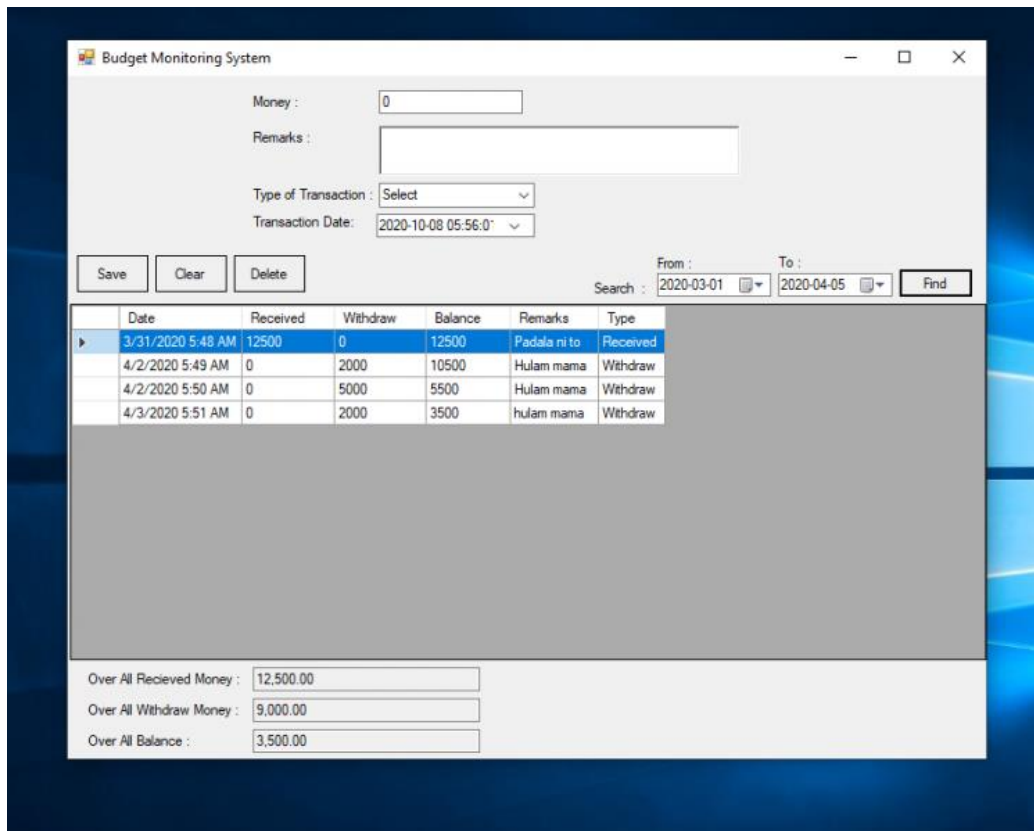


Figure 1. Budget Management System Interface in Visual Basic .NET

A summary panel positioned below the table provides users with real-time financial statistics. It displays three core indicators: Total Income, Total Expenses, and Current Balance, all calculated dynamically through SQL aggregate functions. These values automatically refresh after any data change, offering an immediate overview of financial standing. Visual cues enhance interpretation—positive balances appear in green, signaling surplus, while negative balances appear in red, indicating deficit. The color-coded feedback helps users quickly assess their financial condition without requiring manual calculation. An additional date filtering function enables users to analyze financial data across specific periods. By selecting a start and end date, users can focus on defined timeframes—such as a week, month, or quarter—to evaluate spending and income patterns. The system adjusts both the data grid and the summary panel to reflect filtered results, allowing targeted financial analysis and supporting informed budgeting decisions. Comprehensive testing confirmed the reliability and accuracy of the Budget Management System.

Unit testing verified the correctness of individual components, including income and expense recording, updating, and deletion processes. Integration testing ensured smooth data communication between the Visual Basic .NET interface and the MySQL database, with all SQL operations executing successfully. Functional testing validated that all user-facing features performed as specified—recording, viewing, filtering, updating, and summarizing transactions—without producing logical or runtime errors. Performance testing further demonstrated the system's stability under varying data volumes. Even with databases containing thousands of records, all core operations executed within acceptable timeframes. Transaction insertion, modification, and deletion were typically completed in under half a second, while data retrieval and aggregation queries remained responsive at under one second for databases containing up to 5,000 entries. Usability testing, conducted with participants of diverse technical backgrounds, confirmed that the interface was easy to navigate. Users completed all tasks successfully with minimal guidance and reported that the color-coded balance indicators and minimal design enhanced the overall clarity of the system.

4.2 Result

The evaluation of the Budget Management System highlights several key strengths and practical implications, alongside challenges and areas for future enhancement. The foremost advantage of the system lies in its simplicity and accessibility. Unlike many commercial financial management tools that require continuous internet connectivity or subscriptions, this application offers offline functionality, eliminating dependence on external

servers and recurring costs. This makes it particularly valuable in regions with limited connectivity or for individuals seeking full control over their financial data. Data privacy emerged as a defining characteristic of the system's design philosophy. Because all data is stored locally, users retain exclusive ownership of their information, avoiding the risks associated with cloud-based applications. This feature addresses widespread concerns over data breaches and unauthorized access to personal financial records. Additionally, the open-source nature of the system enhances transparency and customizability. Users with programming knowledge can adapt the application to specific needs—such as integrating new features, adjusting the interface, or connecting to other systems—without relying on proprietary software providers. The system also provides educational and developmental value. For students and novice developers, it serves as a tangible demonstration of software engineering principles, including database integration, interface design, and structured programming in Visual Basic .NET. The availability of well-documented source code enables learners to understand modular code organization, SQL operations, and event-driven programming in a real-world context. For small-scale users—such as freelancers or individuals managing personal budgets—the system offers an effective solution for maintaining accurate financial records without the overhead of professional accounting software. However, several limitations were identified during implementation. The installation process, which requires users to configure MySQL via XAMPP and import database scripts manually, may present challenges for non-technical users. Automating this setup through an installation wizard would simplify deployment and broaden accessibility.

The system's current single-user design restricts collaborative or household budgeting, where shared access is desirable. Future iterations could adopt a multi-user architecture incorporating authentication and role management to enable concurrent usage. Additionally, while the desktop-only version serves the current scope well, cross-platform compatibility—particularly for mobile and macOS environments—would significantly enhance usability and reach. User feedback also revealed potential areas for improvement, including the addition of expense categorization, graphical reporting, and recurring transaction features. Integrating visual analytics, such as pie charts or bar graphs for expense distribution and trend tracking, would allow users to interpret their data more intuitively. Implementing automated backup and export features to CSV, Excel, or PDF formats would enhance data security and interoperability. Multi-currency support and budget planning tools could further extend the application's relevance in diverse financial contexts. From a comparative perspective, the system occupies a unique niche between basic spreadsheet budgeting and complex commercial software such as Mint, Quicken, or YNAB. While commercial solutions offer a wider range of automated features, they often introduce complexity, privacy concerns, and financial barriers. The Budget Management System, in contrast, prioritizes usability, privacy, and control—key attributes for users who value simplicity over automation. Its lightweight design and local data management provide an attractive alternative for individuals seeking autonomy over their financial tracking process. In a broader context, this research reinforces the principle that effective technology does not require unnecessary complexity to deliver value. The system exemplifies user-centered design by addressing real needs—simplicity, control, and affordability—rather than striving for feature abundance. Its successful implementation validates that appropriate technology, when grounded in usability and accessibility, can substantially improve personal financial management practices. Looking ahead, future development may include the creation of a mobile companion app synchronized with the desktop version, automated database configuration, and visual dashboards for financial insight. Each enhancement would build upon the system's solid foundation while preserving its defining characteristics: ease of use, offline operation, and respect for user autonomy. In summary, the Budget Management System fulfills its objective as an accessible and reliable personal finance application. The results confirm that the system performs efficiently, meets all specified requirements, and offers tangible benefits in privacy, usability, and educational utility. While certain technical and usability limitations remain, they present opportunities for meaningful future improvement. Overall, the project demonstrates that simple, well-structured desktop applications can effectively support individual financial literacy and management in an increasingly complex digital economy.

5 | CONCLUSIONS AND RECOMMENDATION

This research successfully designed and implemented a Budget Management System using Visual Basic .NET 2015 and a MySQL database, providing a practical, accessible, and privacy-focused tool for personal financial management. The system effectively fulfills the identified need for an easy-to-use, offline budgeting application that enables users to track income and expenses, review transaction history, and analyze financial behavior in a structured and reliable manner. By applying established software development principles within the Software Development Life Cycle (SDLC) framework, the project demonstrates that efficient and scalable financial management solutions can be built using readily available technologies without imposing licensing restrictions or subscription costs. Functionally, the Budget Management System meets all essential requirements for personal

finance tracking. Users can record, modify, and delete transactions with descriptive remarks that provide contextual clarity, while the system accurately distinguishes between income and expenditure entries. Automatic calculation of total income, total expenses, and current balance provides immediate financial visibility through a clear summary panel enhanced by intuitive color indicators that signal surplus or deficit. The inclusion of date-based filtering and sortable transaction history enables users to analyze specific financial periods and observe spending trends over time. The integration of Visual Basic .NET with MySQL through ADO.NET ensures efficient data management, secure access, and reliable query execution. Additionally, the interface design prioritizes clarity, presenting all major operations within a single main form to minimize navigation and simplify user interaction. Extensive testing validated the system's reliability and performance. Unit and integration tests confirmed accurate interaction between components, functional tests demonstrated compliance with all user requirements, and performance tests verified that the system maintains responsiveness even with large datasets. Usability testing with participants of diverse technical backgrounds confirmed that the interface was intuitive and accessible, requiring minimal technical expertise.

These evaluations affirm the system's robustness, consistency, and readiness for real-world deployment. The system also offers several advantages that contribute to its usability and practicality. Its simplicity ensures accessibility for users with varying technical abilities, while its offline capability eliminates reliance on internet connectivity—an important consideration in regions with unstable access. Local data storage provides users full control over their financial information, addressing concerns about privacy and security in cloud-based applications. The open-source nature of the system enhances cost-effectiveness and encourages collaborative development, allowing students, educators, and developers to adapt the software for educational or practical purposes. The project demonstrates that user-centered design and appropriate technology can deliver impactful solutions without unnecessary complexity. Despite its success, certain limitations were identified. The database configuration process still requires manual setup through XAMPP and phpMyAdmin, which may challenge non-technical users. The application's dependency on Windows restricts cross-platform deployment, while its single-user design limits shared budgeting scenarios such as household or group use. Data backup must also be performed manually, creating potential risks of data loss if users fail to perform regular backups. These limitations, though not diminishing the system's effectiveness, indicate clear directions for improvement. In the broader financial technology landscape, the Budget Management System occupies a position between basic spreadsheet tools and feature-heavy commercial applications such as Mint or YNAB. While commercial systems offer advanced automation and analytics, they often involve costs, privacy risks, and overwhelming complexity. In contrast, the Budget Management System provides a lightweight, autonomous alternative emphasizing privacy, usability, and control—characteristics aligned with the needs of individuals seeking simple yet reliable budgeting tools. The project therefore contributes both practically and academically, offering an operational financial management solution and a structured example of database-driven software development guided by accessibility, transparency, and user empowerment. Building upon the findings and feedback, several recommendations are proposed for further development, deployment, and research to enhance the system's performance and relevance. Future development should prioritize simplifying installation by introducing an automated setup process that configures the database and application environment with minimal user input. Expanding compatibility through .NET Core or Xamarin frameworks would allow deployment across macOS, Linux, and mobile platforms. Adding expense categorization would enable users to classify transactions by type—such as housing, food, or utilities—providing richer financial insights.

Integrating graphical reports such as charts and trend lines would support visual analysis of income and spending patterns. Incorporating budgeting features like spending limits and notifications could promote proactive financial planning, while a synchronized mobile version would extend usability. Automated backups and export functions to formats such as CSV, Excel, or PDF would further strengthen data protection and flexibility. For deployment and community support, developing comprehensive documentation, including installation guides and video tutorials, would improve user adoption. Establishing an online support community could facilitate feedback and collaborative development. Implementing ethical telemetry or usage analytics—with explicit user consent—would provide insights into feature utilization and inform targeted improvements. Partnerships with educational institutions and financial literacy programs could also promote broader adoption and integration into academic contexts. From a research perspective, future studies should examine the system's long-term impact on users' financial behavior and literacy, comparing its effectiveness with commercial alternatives. Investigating factors that influence adoption, usability, and sustained engagement could inform future iterations of the software. Cross-cultural and socio-economic analyses would further evaluate its adaptability in diverse contexts. The potential integration of artificial intelligence or machine learning could also be explored to offer personalized financial insights while preserving the system's privacy-focused architecture. In conclusion, this research achieved its primary objective of developing a functional, user-friendly, and secure Budget Management System that empowers individuals to manage their finances effectively. The system's strength lies in its simplicity, offline functionality, and

adaptability, providing a meaningful alternative to complex commercial tools. The limitations identified do not undermine its practicality but instead point toward opportunities for refinement and expansion. This project exemplifies appropriate technology—solutions designed for usability and inclusivity rather than feature excess. Its open-source nature ensures continued growth through community participation and educational use. Ultimately, the Budget Management System demonstrates that accessible, ethically designed financial applications can significantly contribute to improving financial awareness, literacy, and stability among individuals navigating an increasingly complex economic environment.

REFERENCES

- Ameriks, J., Caplin, A., & Leahy, J. (2003). Wealth accumulation and the propensity to plan. *The Quarterly Journal of Economics*, 118(3), 1007–1047. <https://doi.org/10.1162/00335530360698487>
- Andriana, M., Panjaitan, R., & Sumarlin, T. (2021). Sistem informasi pengelolaan anggaran dengan metode R&D. *Evolusi: Jurnal Sains dan Manajemen*, 9(2). <https://doi.org/10.31294/evolusi.v9i2.11147>
- Fernandes, D., Lynch, J. G., & Netemeyer, R. G. (2014). Financial literacy, financial education, and downstream financial behaviors. *Management Science*, 60(8), 1861–1883. <https://doi.org/10.1287/mnsc.2013.1849>
- Grable, J. E., & Joo, S. H. (2004). Environmental and biopsychosocial factors associated with financial risk tolerance. *Financial Counseling and Planning*, 15(1), 73–82.
- Lusardi, A., & Mitchell, O. S. (2014). The economic importance of financial literacy: Theory and evidence. *Journal of Economic Literature*, 52(1), 5–44. <https://doi.org/10.1257/jel.52.1.5>
- Microsoft Corporation. (2015). *Visual Basic .NET language specification*. Microsoft Developer Network. <https://docs.microsoft.com/en-us/dotnet/visual-basic/>
- Oracle Corporation. (2023). *MySQL 8.0 reference manual*. <https://dev.mysql.com/doc/refman/8.0/en/>
- Patel, N. K., & Patel, R. M. (2016). Development of personal finance management system using .NET framework. *International Journal of Computer Applications*, 145(8), 23–27. <https://doi.org/10.5120/ijca2016910720>
- Ramsey, D. (2013). *The total money makeover: A proven plan for financial fitness*. Thomas Nelson Publishers.
- Riadi, R., Tedja, B., & Basatha, R. (2024). Pembuatan aplikasi toko komputer dengan CRUD berbasis Visual Basic dan MySQL. *Al-Dyas*, 4(1), 411–423. <https://doi.org/10.58578/aldyas.v4i1.4518>
- Setiawan, T. (2025). Sistem informasi penggajian pegawai berbasis desktop pada CV. Hijab Geulis Konveksi. *Internal: Information System Journal*, 8(1), 37–46. <https://doi.org/10.32627/internal.v8i1.1424>
- Shim, S., Barber, B. L., Card, N. A., Xiao, J. J., & Serido, J. (2010). Financial socialization of first-year college students: The roles of parents, work, and education. *Journal of Youth and Adolescence*, 39(12), 1457–1470. <https://doi.org/10.1007/s10964-009-9432-x>
- Tafonao, R., Hondro, R., & Panjaitan, M. (2023). Perancangan sistem informasi anggaran pendapatan belanja daerah kecamatan dengan metode line-item budgeting. *Bulletin of Information and Data Science (Bulletinds)*, 2(3), 118–123. <https://doi.org/10.47065/bulletinds.v2i3.5312>
- Thaler, R. H., & Sunstein, C. R. (2008). *Nudge: Improving decisions about health, wealth, and happiness*. Yale University Press.
- Troelsen, A., & Japikse, P. (2017). *Pro Visual Basic 2015*. Apress.
- Ultariani, N., Putra, N., & Amroni, A. (2020). Perancangan sistem informasi persediaan dan penjualan pada Toko Ria Bangunan dengan menggunakan bahasa pemrograman Visual Basic 2010 dan database MySQL. *Jurnal Digit*, 10(2), 220. <https://doi.org/10.51920/jd.v10i2.172>

- Xiao, J. J., & O'Neill, B. (2016). Consumer financial education and financial capability. *International Journal of Consumer Studies*, 40(6), 712–721. <https://doi.org/10.1111/ijcs.12285>
- Zaltin, A. (2025). Sistem informasi penerimaan dan pengeluaran kas berbasis desktop. *RJTI: Rekayasa Jaringan dan Teknologi Informasi*, 4(2). <https://doi.org/10.30606/rjti.v4i2.3447>

How to cite this article: Janobe, J. (2025). Development of Budget Management System Using Visual Basic .NET and MySQL Database: A Desktop Application for Personal Financial Tracking. *Journal Dekstop Application (JDA)*, 4(1). <https://doi.org/10.59431/jda.v4i1.660>.