

Personal Financial Tracker App

Benjamin Etheridge
Supervised By: Dr Donggyun Han

Department of Computer Science
Royal Holloway, University of London

Abstract

Managing personal finances can be a daunting task for many, leading to financial stress and poor decision-making. In the U.K., 47% of adults do not feel confident in financial products and services[5]. This lack of confidence is partially due to financial illiteracy and the overcomplication of financial data, which contributes to poor understanding of personal finances. For instance, 22% of individuals have less than £100 in savings, and 47% have no financial goals for the next five years [5].

Studies have shown that visual information is retained more effectively than written or verbal data [3]. This project will focus on translating financial information into visual representations, such as graphical interfaces, while studying the necessary tools to do so. The emphasis will be on mapping data onto graphs that are both visually appealing and easy to understand. React is the chosen frontend framework due to its reusable components, flexibility, and fast rendering capabilities with its Virtual DOM [2]. Additionally, it has Recharts as its own graph library and will be utilized for data visualization, allowing for seamless integration.

Applying N-tier architecture will enable the application to handle large and complex requests while providing scalability [4]. The application will feature a user-friendly interface, ensure a responsive user experience, and efficiently manage data. Essential features will include user authentication, transaction categorization, budget setting, and various visual analytics. Given the sensitivity of the data, encryption will be crucial. Being able to ensure a HTTPS or SSL/TLS certificate for encryption in transit, and potentially encrypting at rest [1], would have lots of benefits. However, having the ability to get a certificate will be rather difficult. Certificates such as these would be essential if my project is to be deployed as they instill trust in the application for the user, giving some form of proof that their data will be secure. And so, these measures will have to be investigated in great depth, to see if it is possible to get them without incurring costs. On top of protecting data, I will also have to ensure methods to prevent attacks on the application. Attacks such as an SQL injection could easily interrupt my website, as many transactions from the front and back end will involve an SQL query. Especially when importing bank statements and transactions, the application will be vulnerable. Prepared statements and not allowing special characters upon import will be some methods taken to hinder potential attacks.

This project will adhere to best practices in software engineering, with an emphasis on modularity, maintainability, and especially security. By integrating state-of-the-art digital solutions with targeted financial education, individuals will be better equipped to navigate their financial landscapes and improve their overall financial well-being.

1 Timeline

1.1 Term 1

- Week 1/2: User login and authentication
- Week 3: Welcome header, dashboard page, link to other pages
- Week 4: Create database for storing data; API for uploading to the database
- Week 5/6: Uploading transactions; import bank statements (CSV in correct format)
- Week 7: Retrieve transactions from the database and format for the webpage
- Week 8/9: Filter functionality to sort transactions by categories
- Week 10: Interim reports
- Week 11: Finalize reports and presentation

1.2 Term 2

- Week 1: Insight page/tab
- Week 2/3: Graphical representations of expenditures over time (by category)
- Week 4/5: Set budgets page
- Week 6: Graphical representation of budget vs. expenditures
- Week 7: Polish user interface
- Week 8: Testing phase
- Week 9: Buffer week
- Week 10/11: Final report and presentation

2 Risks and Mitigations

2.1 Security Risk

When developing an application that allows users to upload bank statements, it is essential to encrypt sensitive data during transmission and at rest. Getting SSL(TLS)/HTTPS certificates would be crucial. Implementing security measures against SQL injection is also necessary, including methods like strict access controls and prepared statements. Poor implementation of real and effective security features can lead to the downfall of the project.

2.2 Time Allocation Risk

Allocating sufficient time for each project phase is crucial to ensure timely progression. If any stage falls behind, it may hinder the ability to deliver the required outcomes. Proper time management and allocation should be enforced.

2.3 Data Storage Considerations

Storing large datasets can be resource-intensive. The application will require a scalable solution with potentially 24-hour access, such as cloud storage services (e.g., AWS S3). Ensuring these services implement robust security measures is essential.

2.4 Design Risks

Users, particularly older individuals, may struggle to submit statements in the required format, leading to errors and frustration. Providing a generalized guide outlining acceptable formats, along with step-by-step instructions and validation checks, can mitigate this risk.

2.5 Coding Risks

Over-reliance on pre-made packages and APIs for functionalities, such as data visualization, may lead to issues if those libraries lack necessary features. It is important to invest time in evaluating libraries to ensure their capabilities and limitations are well understood.

2.6 Financial Cost

Typical methods of protecting applications data online usually require certificates, such as HTTPS or SSL/TLS. Most services that provide these certificates not only have prerequisites to getting them but also incur a fee for the service. If this fee turns out too great to proceed, then customers will have no indication if their data will be protected at all, meaning my application is highly unlikely to be used.

3 References

References

- [1] Catalyst2. Encryption at rest: Its purpose and benefits. <https://www.catalyst2.com/blog/encryption-at-rest-its-purpose-and-benefits-setbacks/>.
- [2] Giuseppe Psaila. Virtual dom: An efficient virtual memory representation for large xml documents, 10 2008.
- [3] R. Moreno R.E. Mayer. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, 38(1):43–52, 2003.
- [4] T. Hazim H. Alrikabi K. Nasser S. Alseelawi, K. Adnan. Design and implementation of an e-learning platform using n-tier architecture. *International Association of Online Engineering*, 2020. <https://www.learntechlib.org/p/216473/>.
- [5] Money Advice Service. Financial capability survey, 2018. [data collection]. 2nd edition, 2018.