

Module No:	COMP6013	Module title:	BSc Computing Project
Degree Programme :	Bachelors		
Project title :	Expense Tracking Application		
Supervisor :	Alexander Rast		
Due date and time:	20/04/23 13:00pm		
Estimated total time to be spent on assignment:		90 hours per student	
Student No:	Student Name:		
19025241	Ali Khan		

Statement of Compliance (please tick to sign)



I declare that the work submitted is my own and that the work I submit is fully in accordance with the University regulations regarding assessments
www.brookes.ac.uk/uniregulations/current

Regulations governing the deposit and use of Oxford Brookes University Projects and Dissertations

Copies of projects/dissertations, submitted in fulfilment of Modular Programme requirements and achieving marks of 60% or above, shall normally be kept by the Library.

Statement of Permission (please tick to sign)



I agree that this dissertation may be available for reading and photocopying in accordance with the Regulations governing use of the Library.

Contents

0: Acknowledgments:	3
1: Abstract:.....	3
2: Introduction:	4
2.1: Background:.....	4
2.2: Aim:	5
2.3: Objectives:	5
2.4: Scope:	6
2.5: Audience:	6
3: Background Review:.....	6
3.1 Existing Expense Tracking Applications	6
3.1.1 Mint:	7
3.1.2 YNAB (You Need A Budget).....	7
3.1.3 Wally	8
3.2 Web Application Development Approaches:.....	8
3.3 Literature Review	8
4 Methodology.....	10
4.1 Design Rationale.....	10
4.2 Functional and Non-functional Requirements	11
4.3 Development Approach.....	12
4.4 Design Specification	13
4.4.1 Coding Language, frameworks, and libraries	14
4.4.2 Front-End	14
4.4.3 Back-end.....	15
4.4.4 Summary of Technologies used	16
5 Results	17
5.1 Walkthrough.....	17
5.2 Testing	28
6 Project Management.....	32
6.1 Professional Issues	32
6.2 Risk Assessment and Mitigation.....	32
6.3 Project Timeline.....	33
7 Conclusion.....	34
7.1 Future plans.....	34
7.2 Self reflection.....	34
Appendices:.....	38

Link to poster:.....	38
Link to video recording:.....	38
Link to source code:	38

EXPENSE TRACKING APPLICATION

0: Acknowledgments:

During this project, there was many difficulties faced and almost impossible to overcome these on my own. I would really like to thank my supervisor Alexander Rast for helping me overcome these issues and allowing me to complete this project.

1: Abstract:

Personal Finance Management is an integral part of maintaining financial stability and achieving long-term goals. Expense tracking applications have gained popularity in recent years as they are great tools to allow individuals to manage their finances more effectively. This project explores the development of an expense tracker application in a web application environment, aiming to provide user-friendly and effective solutions for managing personal finances.

This web application offers features such as inputting expenses, categorising transactions, setting budgets, and visualizing financial data through charts and graphs. These functionalities allow users to gain insights into their spending habits and allow them to make necessary adjustments to achieve their financial goals.

This report starts off with an extensive background review, examining existing expense tracking applications, personal finance techniques and web application development approaches. Gathering this information laid the foundation for the web application's design, development, and implementation. The methodology section outlines the approach and technology used, including the plans, testing strategies, and version control and management.

The results section shows findings from the usability testing, performance evaluation and user feedback, providing insight into the web application's effectiveness and identifying key areas for improvement. This report concludes with a reflection on the project's successes and challenges faced allowing for improvement for future development.

This project contributes towards the ongoing development of effective financial management tools and promotes better personal finance management habits. With a concise outline of the projects background, aims, and achievements, this abstract summarizes the key aspects of this expense tracking web application.

2: Introduction:

2.1: Background:

Managing personal finances effectively is important for individuals and families in today's economy. With rapid growth of financial technology (fintech) and wide use of smartphones, there has been a rise of mobile and web applications to address the problem for the need of tracking and managing expenses. Many of these applications offer different ranges of functionalities varying from simple expense tracking to more advanced analysis. However, there is no in-between when it comes to the to cater to user's needs. As many users have become normalised to personalized experiences in digital products, there is an increasing demand for expense tracking solutions that provide tailored insights to users based on spending habits and financial goals.

Some issues occur when finding the right expense tracker to use as many of them are mainly targeted to professionals such as accountants or financial advisors, this approach can leave regular users confused with all the advanced tools and terminology, making it more challenging to use the application. Furthermore, leaving a huge learning gap when a user signs up to a new platform. Normal everyday users may not be able to carve out enough time during their day to learn advanced tools, so it is important for them to be able to use an application which is straightforward and requires little or no learning. In this current market for financial expense tracking, there are many options to choose from ranging from free to use applications and monthly subscriptions. This market is vast and contains a range of expense and budget tracking applications however it is all up to the user which is right for them respective to the budget methods that the applications use [1]. There is one main competitor in this market, which is Mint, they are a free to use software which is easy to use however they do have some learning curves which come with their navigation. Moreover, other applications are either too complex for users or charge their users a monthly subscription which means that users will be forced to pay every month as all their finances are on that specified application. I believe that this industry has a gap when it comes to the simplified expense tracker approach as there is only 1 main competitor and improvements are possible in which that they lack. A more simplified approach to an expense tracker is ideal to ensure that users understand the application's features and feel confident navigating it. In terms of the human computer interaction (HCI) study, it is extremely important to build an application which would follow everyday users' needs so that the relationship between the user and the application is established [3]. Implementing a user-centered design is key for the achieving the best possible results.

With the rise of cost of living, users may find that their expenses will increase overtime due to the rise of groceries or gas however with use of tracking their expenses they are able to check-in every now and again to make sure their spending plan is up to date and on track toward their financial goals [2]. Staying up to date with finances is crucial as it reduces risk of overspending and allows you to be able to save your money. Being able to track everyday expenses allows for sustainable growth financially and gives users the ability to structure their outgoings and savings accordingly.

This topic I have chosen is somewhat relatable to me, having many troubles with financial management throughout university it was apparent that I had not managed my money well. This had many drawbacks not being able to go out as often and many plans being cancelled due to poor money management. These issues could have been avoided if I had planned my expenses and budgeted them accordingly through an application. Which in course would've allowed for better money management and potentially savings and less spending on unnecessary things. Moreover, showing that the need for an expense tracker application is needed and must serve the users in a correct way. Simplicity and a user-centric nature will be important as time is crucial while studying and users won't be able to learn complex applications alongside studying.

Technologies used in development for such application is important as it must be mindful of user experience and not allow for any security risks. As expense tracker's contain financial data for individuals it is essential not to allow any security issues when developing, this can be prevented by having the correct architecture and backend. Many app development organisations use SaaS (Software as a service) Solutions to develop the back end using pre-configured modules [4]. This could be a great thing to use while development however could pose security risks as if the software in which that you are using in development has underlying security risks, then that allows your application to adopt those security risks. This is the reason I have chosen to create my backend from scratch using known JavaScript frameworks like express and node.js. This allows minimal security risks if I don't include any in the code. Furthermore, for the front end, it has become apparent that many companies opt for HTML and CSS, in which I agree to some extent I believe that another JavaScript library like react is perfect to use as it includes many other modules to improve the UI (User Interface) and UX (User Experience). Additionally, allowing for improved usability and responsiveness per device.

2.2: Aim:

The aim of this project is to develop a user-friendly web application for personal expense tracking and management tool that assists users in organising their expenses, identifying spending patterns, setting budgets, and providing analysis tools to enable better informed financial decisions.

2.3: Objectives:

1: Conduct a literature review on existing expense tracking applications, personal finance techniques and different types of web application development approaches.

- 2: Design a user-centric application interface which prioritizes usability, simplicity, and accessibility so that users can easily navigate and understand the application's features.
- 3: Develop the application using modern technologies such as Express and Node.js for the backend and React and Material-UI for the front end.
- 4: Implement features which allows users to input expenses, categorise transactions, set budgets, and visualize their data through charts and graphs.
- 5: Test the application using Unit testing, Integration testing, acceptance testing [5] and usability testing.
- 6: Evaluate the software developed and documenting the process, including the design, development, and testing stages. This will provide an understanding of the projects successes and challenges faced, offering guidance for future improvement and development.

2.4: Scope:

This project will focus on 3 primary aspects: Firstly, designing a user-friendly interface for inputting, categorising, and visualising personal expenses. Secondly, implementing effective budgeting methods, visualization of different analysis types for expenses and budgeting to help users achieve their financial goals. Lastly, ensuring that the application is secure and accessible, offering simplicity and consistency in user experience.

2.5: Audience:

The main target audience for this project would be millennial spenders and families seeking a straightforward and effective solution for managing their finances. "Ninety-two million millennials will soon to be in their prime spending years, they command \$1.3 trillion in annual spending." [6] Increasing demand for expense tracking and personal financial management tools is on a rise and the expense tracking web application aims to cater to users with varying financial background and knowledge. Additionally, this project may serve as a building block or inspiration for future developers looking to build a more user-centric, simple financial management tool.

3: Background Review:

This topic of my dissertation presents a themed literature review on existing approaches to an expense tracking application. This review also critically appraises various sources, comparing features, functionalities, and the different limitations of existing applications, as well as discussing personal finance techniques and existing web application development approaches.

3.1 Existing Expense Tracking Applications:

This section includes a critical appraisal of existing expense tracking applications, mainly focusing on features, functionalities, and limitations. The applications that will be reviewed in this section are Mint, YNAB (You Need A Budget) and Wally.

3.1.1 Mint:

Mint is a popular personal finance management tool that offers many features such as expense tracking, budgeting, bill reminders and credit score monitoring [7]. Mint allows the users to link their bank accounts and credit cards, providing real-time information and updates on their transactions and account balances. Mint includes a few strengths; firstly, automatically categorising transactions and providing insights into spending patterns. Secondly, personalised customization features allowing users to add tags and reorganise transactions whenever necessary. Thirdly, strong security features allow mint to gain a reputation by using multi-factor authentication, strong encryption and allowing users to be able to control their mobile access remotely in a case of them losing their mobile phone [8]. The limitations of this application are that there are often connectivity issues with user's accounts which can be annoying at times as users may be needing to get an update on their finances in an urgent situation or causing them to reschedule their time. Furthermore, another disadvantage could be the advertisements, this reduces the user-centric nature of the web application as users wouldn't want to see adverts popping up or around the page as they may accidentally click on them. This takes away from their user experience however as a free application they do have to make money to fund this so it is understandable why they would have adverts.

3.1.2 YNAB (You Need A Budget)

YNAB is a comprehensive budgeting application which focuses on the zero-based budgeting technique, this is where users allocate every pound or in this case dollar into a specific category [9]. This application really emphasises the importance of allocating funds to saving goals and living on the previous month's income. YNAB offers many features such as real-time syncing with bank accounts, customisable categories, and an outside the box way of thinking into financial planning [10]. Along with all the great features that this application has to offer it also has a few limitations; firstly, being the price, they offer a 34-day free trial which is great for users to test out the application however they do charge \$14.99 which may not apply to some users as some of the competitors such as mint are free of charge. Secondly, another limitation is the user usability, when navigating the application users may feel overwhelmed at first as it takes a while to learn how to use the application which isn't great because you are paying for a premium service and then you will have to learn which most likely will put users off. Thirdly, YNAB doesn't have many features as some of their competing free apps, such as there is no bill payment alerts, tracking of investments and this app solely focuses on budgeting. This application stands out for its approach to money management however its usability lets it down, for example, users have stated that they were using a google spreadsheet and after trying out YNAB they'd rather stay with the spreadsheet as they felt they were getting the same features and its much simpler to use.

3.1.3 Wally

Wally is a user-friendly expense tracking application that offers features such as budgeting, expense categorisation and receipt scanning [11]. This applications main attraction is its simplicity, mainly focusing on providing users with an easy-to-use platform to manage finances. However, wally lacks advanced features which are found in their competitor's applications such as automatic transaction importing and detailed financial analysis. Also, another limitation on this app is that they charge in-app purchases for add-ons starting at \$1.99 which may put some users off from using this app [12].

3.2 Web Application Development Approaches:

There are various approaches for web application development that can be adopted when creating an expense tracking application. Popular technologies and frameworks for building such applications include Express and Node.js for backend development and React and Material-UI for frontend development [13]. This is based on preference so I believe that this source or any source that will be used for web development approaches will not be factual however give a great insight into multiple different options. There also are many different programming stacks to build applications in a structured way, such as Ruby on Rails (RoR), LAMP (Linux, Apache, MySql, Php) and MERN (MongoDB, Express, React, Node.js). There are many benefits and drawbacks when using a specific stack such as a drawback could be; time consuming to learn and implement, which would be the same for most stacks unless you are familiar with the coding language. Furthermore, some stacks may involve security vulnerabilities if you are unable to use proper implementation of code and structuring code accordingly. However, there are benefits for using a stack such as the asynchronous style of coding as you know that they are able to work together in a correct way, however it does come back to the point on learning how to do so.

3.3 Literature Review

Aggarwal, S. (2018). Modern Web-Development using ReactJS [14]

This piece of literature describes the process in which ReactJS can be used in modern day web development explaining the features, working principles, differences, and limitations of ReactJS in comparison to other web development frameworks. This article highlights many features of ReactJS, such as its lightweight DOM (document object model) for better performance, easy learning curve, JSX language, one-way data flow, and virtual DOM.

Aggarwal explains how using a virtual DOM improves the performance of web applications. This method involves creating a virtual representation of the browser DOM in memory, comparing it to the actual DOM, and only updating the necessary nodes. This approach minimizes the DOM manipulation which results in better performance for web applications.

The author also examines the simplicity of ReactJS, noting that it has an easy learning curve and that the JSX syntax makes it accessible for developers. JSX, which is an XML-like language, allows developers to write markup for components more easily, providing a more natural experience.

Aggarwal also acknowledges that ReactJS has some limitations, such as only handling the View entity in the Model-View-Controller (MVC) architecture, requiring a few additional tools to complete development for a full project. Additionally, some developers may find that the use of inline templates and JSX very complex and tiring. Lastly, ReactJS can compile time errors instead of runtime errors which many developers may find frustrating.

Other than having a few limitations, Aggarwal concludes that ReactJS is a game changer in the field of web development, offering the features needed to meet the demands of modern user experience design trends. ReactJS's ability to create efficient, and dynamic web applications compliments it by positioning it as a significant use for future web development.

Turcomat.org. (2023). *View of Spending Tracker: A Smart Approach to Track Daily Expense*. [15]

This piece of literature presents a daily expense management system, designed to efficiently track day-to-day expenses. The application aims to help users track their daily expenses and maintain records of the transactions through a computerized system, eliminating the need for a hardcopy. This system is developed in java (Apache Netbeans 11.3) and MySQL Workbench and is designed as a GUI-based application.

The authors provide a comparative analysis of traditional and technical approaches for tracking expenses and budget performance. Discussing proposed architecture which includes a database schema, data dictionary and a UML class diagram. The article shows the application's various features such as the main window, add category, category box, remove button, add category window, calendar, view expense (day wise/category-wise) window, menu option, search button and refresh button.

The implementation process of this Spending Tracker app is described using a waterfall model, which is a linear process model where each main activity of an operation is represented as a different phase. The authors discuss various modules,

such as the welcome note module, main window module, add amount, delete expense, view expense, add module, categories module, and search module.

The authors believe that the Spending Tracker app will help users manage their daily expenses, guide, and inform them of their spending habits, and contribute to better financial management. This application will do what is stated above on the other hand I believe that user-friendly nature of this application lets it down, mainly with the GUI. The authors have signified making the expense tracking process more accessible and engaging for users, however I don't believe that this application as shown in the images is very user appealing as the colours are very mixed with no set theme, also looking confusing to navigate. In this literature stating a user experience is crucial in users to consistently use the application which isn't true at all in my opinion.

Overall, the article provides a comprehensive overview of the Spending Tracker application, its features, and the proposed architecture. The authors have successfully demonstrated how the application can be helpful in managing daily expenses and guiding users towards better financial management. However, this literature piece could benefit from a more in-depth discussion of the applications performance, user feedback and potential limitations.

4 Methodology

4.1 Design Rationale

Within this design rationale it is crucial to ensure that the considerations made lead to the correct technology stack that would benefit this project from beginning to end. The choice of technology chosen is mainly driven by the insights into development gained by research and the literature review. Various factors had influenced the decision to go a certain way for the development process.

The literature review played a critical part in identifying key frameworks, libraries, and coding languages to use within this project. It had offered key information outlining the strengths, weaknesses, and potential for what the technology could become. This was well informing, allowing a decision to be made using the existing knowledge of techniques in modern- day web development.

Many top-level philosophies help guide the technology selection process. These philosophies show the importance of taking a logical development process. Some of these philosophies are user-centric design, maintainability, scalability, and modularity. Additionally, the need for an adaptable, high performing, and secure technology is also key for the implementation of the application's functional and non-functional requirements.

The main motivation to choosing a specific technology stack was based on many different considerations. Some of these key factors include;

- Compatibility: The technologies that are chosen should work well together, ensuring effective integration between the frontend and backend.
- Community: Should have a strong community of developers which allows for extensive resources, further knowledge and support which is very important during the development process.
- Learning curve: The Technology should be accessible and relatively easy to learn, which enables an effective development process.
- Performance: The performance in the technology needs to be fast, contributing to a more responsive application and increasing the user experience.
- Scalability: The technology should allow for expansion for future use. Also being adaptable enables the applications relevancy to increase the requirements.

Considering these multiple factors and the insights gained through research and literature, the decision to use JavaScript for the main coding language, alongside React.js for frontend development and Express.js and MongoDB for backend development, was made. This technology stack offers a great balance for functionalities, high performance, and ease of use. Within the subsequent sections, functional and non-functional requirements, development approach and design specifications will be discussed in further detail.

4.2 Functional and Non-functional Requirements

Functional Requirements:

User Authentication:

- User registration with email and password
- User Log in
- User Log out

Budget Management:

- Create a budget for a specific category
- View a list of budgets
- Edit a budget
- Delete a budget
- Display insights for the budget:
 - Difference between budget and expense amount
 - Average monthly expenses
 - How much of the budget percentage is spent
 - A progress visually showing how much of the budget is left
 - Total Budget and remaining budget

Expense Management:

- Create an expense with a category, description, amount and date.
- View a list of expenses
- Edit an expense
- Delete an expense
- Filter expenses by category and date range

Data Visualisation:

- Display expenses by category choice
- Allow choices of charts to be shown (Pie chart, Bar chart, stacked bar chart, line graph)
- Display charts according to the date range chosen.

Non-Functional Requirements:

Usability:

- Simple and intuitive user interface
- Clear and easily understandable navigation
- Application should work well on different devices and screen sizes

Performance:

- Fast loading and response time for frontend components/pages
- Efficient handling of data fetching and manipulation

Scalability:

- Frontend and backend architecture supportable for future expansion

Security:

- Secure user authentication and data storage
- Proper measures need to be taken to prevent unauthorized access and data leaks

4.3 Development Approach

This project will use the waterfall development methodology. The waterfall approach is a linear and more traditional approach to the development process. The progress flows through a sequential way collecting and documenting requirements, Design, implementation, testing and then maintenance [16]. The waterfall model is shown below:

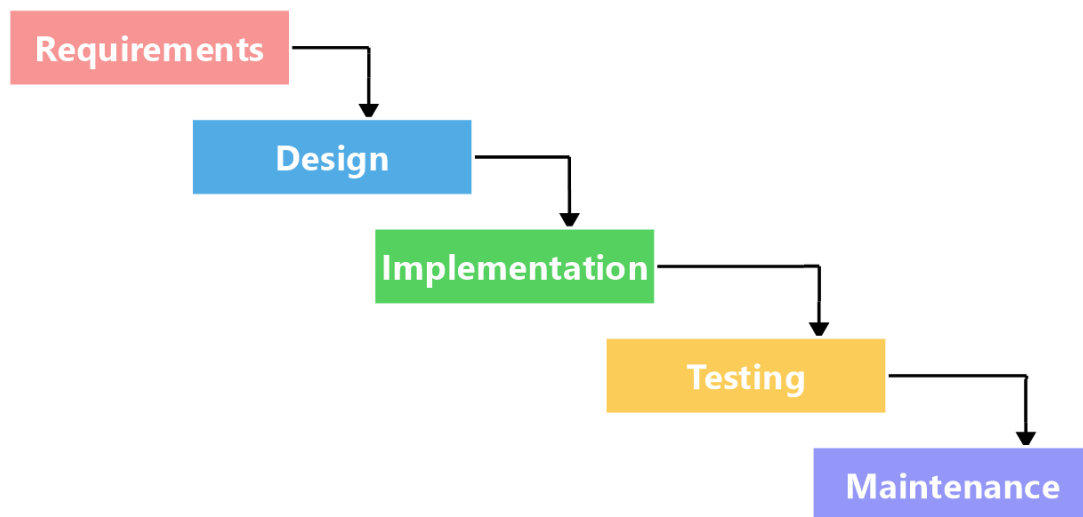


Figure 1 Waterfall Model

Furthermore, comparing to Agile methodologies, I believe that the waterfall approach has a more structured approach to the development process, with having clearly defined stages and deliverables from the beginning. On the other hand, agile allows for flexibility throughout the development process which can cause for mistakes and altering from the task at hand. The waterfall methodology is more suitable for projects with pre-determined requirements and having a limited scope for many changes during the development process.

The main advantage of using the waterfall methodology is that it is simple, making it clear and easy to understand and follow through. Additionally, this approach emphasizes the documentation of development which can be helpful for maintenance and the overall understanding of the system.

Within this project I believe that the waterfall approach is more appropriate due to the clear vision and understanding throughout the development process. Moreover, the structuring of this methodology will aid in effective project management and progress tracking.

To ensure that I am successful in using the waterfall methodology, it is important to use a lot of time and effort in the initial stages of gathering requirements and analysis, as well as the design of the system and software. Setting a clear scope and initialising a step-to-step process which allows me to follow through with my plan with accuracy capitalising on what I may be familiar with and knowing what I would need to learn along the journey. Documentation will be maintained throughout the project to allow for smooth project development and allow for effective troubleshooting in the case I run into troubles when in the maintenance stages of this methodology.

4.4 Design Specification

This section will outline the technical stack and components that have been used within the project to meet all functional requirements.

4.4.1 Coding Language, frameworks, and libraries

This project and study is carried out using JavaScript as the main and only language. There are many benefits for mainly using one language as it de intensifies the development process of the project which allows for less roadblocks and errors occurring. Furthermore, all the development is going to be through visual studio code which caters for all language types however in this case JavaScript. Other languages such as python was considered for server-side programming due to its ability for design implementation and speed. However, JavaScript was selected as already being somewhat relevant to the coding language also having a strong community of online users and support. However, picking JavaScript wasn't all positives; for example running into run time errors and issues with data leaks within the react-dom module in node.js(EXPLAIN LATER). JavaScript's speed and effectiveness when it comes to building a web-based application is second to none and allows for relatively easy expansion with of course many learning curves.

All the dependencies which were added and managed throughout this project were handled using NPM (node package manager). This is extremely powerful and easy to use and is included with the installation of node.js. Running commands such as "npm install express mongoose cors" which essentially installs the necessary packages for Express and Mongodb. This is one of many examples of how simple it is to use. All the frameworks, libraries and packages were installed within the project folder, specifically within the node modules folder which contains all of the packages and tools necessary for development with JavaScript. Furthermore, all packages are defined within the .json file for future reference while development and allowing any version changes due to incompatibility reasons.

Overall, within this project there are important libraries that need to be installed to allow this project to have an overall structure and any development integrity. This project needs to have a solid backend and front-end functionality allowing them both to converse with one another to make this application come to fruition using Nodejs as the main tool for allowing this to happen.

4.4.2 Front-End

While deciding which framework to use for the frontend I had many options within the language of JavaScript. First main choice was react, this is due to its modern day popularity within front end development, allowing for reusable components, softer learning curve and the ability to write components without the classes. It is known that nearly 40% of developers worldwide use react and love it [20]. Furthermore, this was the main choice for this project due to being able to write JSX which allows writing html snippets inside JavaScript files, allowing for working harmoniously with state management, routing and API integrations. Moreover, Reacts library uses a virtual DOM (Document Object Model) by creating an in-memory cache. This compares the previous and subsequent states of the components whereas the real DOM renders the components states that were changed. Additionally, React uses many popular components such as Material UI. This component allows for different web components to be leveraged into making an elegant UI design. It has many

different features with styling, creating a layout, forms, navigation, and many other widgets.

The second choice that was chosen for this project was Angular. This is a Typescript based open source application framework developed by Google. The version 2 of this framework introduced many features such as dependency injection, asynchronous complications etc. This is very popular amongst developers according to a survey which was taken nearly 60% of developers had hands-on experience with Angular [20]. It is a fully-fledged MVC (Model, View, Controller) framework allowing for creation of a well structured application. The main features in which this framework allows for is bi-directional data flow, enabling Angular to be able to connect the DOM to the model through the controller. Using the bi-directional dataflow, you can listen to events and update the data simultaneously between the parent and child components. The hierarchical dependency injection feature makes all the code components testable and reusable, also debugging is less painful offering instant error detection with valuable feedback options.

Out of these two brilliant frameworks, outweighing the benefits and real use cases for this project it made the most sense to move towards React.js as it offers more features which can be taken advantage of. This allows for a well-structured application with a wide range of options in respects to front end development. Additionally, for the project, React is the best choice in regards to following the functionality implementation through the use of additional react components such as React-DOM, Material-UI, React-ChartJS, and most importantly Axios. React-DOM is used to update components when changes have been made which allows for fast and effective UI rendering. Material-UI allows for multiple web developments to be manipulated into making a great user-centric design. React-ChartJS is used to meet the visual analysis functionality by creating and inserting data into multiple types of charts. Lastly, Axios is used to make private API routes within the project in which communicating to the backend for server side development. All of these components of React and Axios being a separate library used is a great front end development structure for implementation and following the functionalities listed in 4.1 of this report.

4.4.3 Back-end

The backend development technologies that were chosen for this project consists of Express.js and MongoDB. This combination for the backend structure is strong due to its high performance and flexibility. There were other choices considered when formulating ideas for the backend, these include Flask and MySQL. Flask uses python which is a completely different language which would come with more learning curves while in development stages. Flask is a strong backend structure to use however it would've been difficult in implementation as using python would've come with multiple learning. Furthermore, MySQL is a familiar choice with many developers when creating web applications, however I believe that MySQL comes with many constraints including the relational data structure of the database design. This is the reasoning I had chosen MongoDB as it takes different data input. MongoDB is a powerful, flexible, and scalable NoSQL database which stores a JSON-type format called BSON (Binary JSON). This is more flexible when it comes to handling large amounts of unstructured data and allowing for the creation of a

Schema-less database, evolvable overtime as new data fields are added, or existing ones are updated. This is more beneficial for this project as it will enable easy adaptation to the multiple functionalities listed and future additions. Additionally, MongoDB's built-in support for horizontal scaling makes it a great choice for building an effective backend infrastructure.

Express.js is very fast and a minimalistic framework built upon Node.js, simplifying the approach of building an adaptable server-side for this project. Providing many features such as routing, middleware and API creation. The API in this project is passed along using Axios from the front-end so the server will receive information to do backend processes. Additionally, another benefit of using Express.js alongside Node.js is the increased performance and responsiveness allowing asynchronous execution when developing, also layering above the user-centric features included in this project. The use of Express will allow the development process to be smooth and facilitate the main functionalities to work effectively.

By using the combination of Express.js and MongoDB, the backend for the expense tracking application will be built on a solid foundation, supporting high performance and scalability. These two frameworks work very seamlessly together passing information to one another through Axios's communication to Express which allows for great data integration and data protection. The security features of using these both together ensures secure communication without data leaks which can improve trustworthiness for the application.

4.4.4 Summary of Technologies used

Frontend:

- React.js: Chosen for popularity, ease of use and support for reusable components allowing for straightforward integration with state management, routing, and API integration.
- Material-UI: component library from React which allows for creation of responsive UI designs with multiple design features for styling, layouts, forms and navigation.
- React-ChartJS: used to create and insert various charts for visual analysis.
- Axios: A library used for making API requests to the backend enabling effective communication between the frontend and backend.

Backend:

- Express.js: Fast, minimalistic web framework built on Node.js, simplifying server-side programming with many features like routing, middleware and API creation.
- MongoDB: Powerful, flexible and adaptable NoSQL database chosen for its ability of storing large unstructured datasets in BSON (Binary JSON) format. Supporting Schema-less design which allows for future expansion of the application.

Language and Tools:

- JavaScript: The main coding language used both in the frontend and backend, allowing for an effective development process and relatively easy error fixing.
- Node.js: A JavaScript runtime which enables the server-side programming, providing the same language across the entire application.
- NPM (Node Package Manger): A tool which is used to manage the dependencies and packages, enabling simple installation and management of the libraries and frameworks used.

This chosen technology is well suited towards this expense tracking application and ensuring a solid foundation for development. The combination of these technologies enables high performance, strong security and opens future development. Selecting and utilizing this technology stack is a great and a valuable learning experience, showing how each piece of technology is used and their synergy to create this project.

5 Results

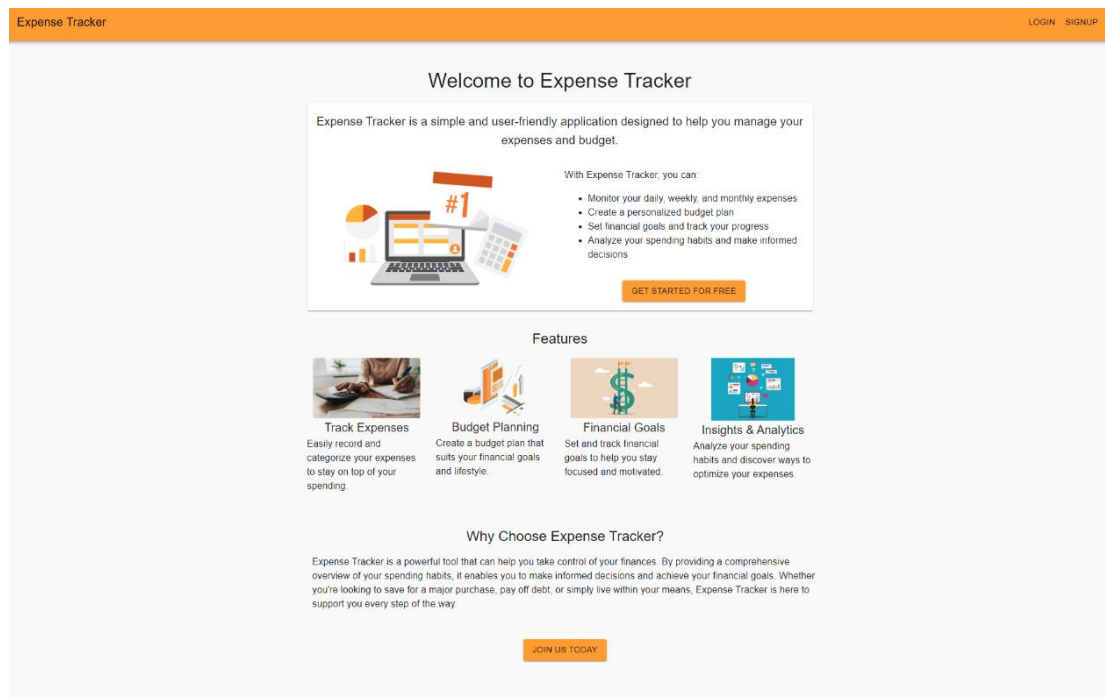
In this section, we will discuss the results of the implementation within the web application. This will mainly focus on the functionality, performance and the consistency following the outlined functional and non-functional requirements. A walkthrough of the application will be provided and explained, highlighting the key features, and demonstrating the effectiveness of following the project's goals. Furthermore, the testing will be detailed, showcasing the test data, and examining the application's ability to meet the established requirements. This extensive analysis will allow us to assess the overall successes and downfalls during the development process.

5.1 Walkthrough

Within this section we will discuss the applications process for users' ability and how well it expresses at outlining the functionality of the project stated in 4.2.

To begin with the MongoDB server will need to be running to allow data to be passed to the database. Frontend will be ran locally using localhost:3000 and the backend server will be running listening to localhost:5000.

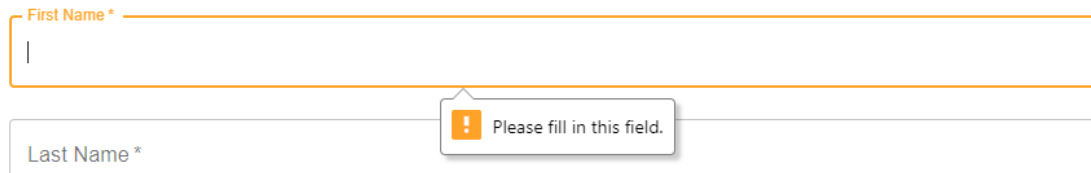
5.1.1 Homepage:



Once the Application is running you will be greeted with the Homepage, This includes context for the user that first enters the site explaining what it is and how it can be used to assist in their expense management. Overall, the design for this page is simple and straight to the point, not going too in depth in explanation and keeping it user friendly. The theme is orange and black as you can see which is neutral keeping the simple theme and allowing for user-centric nature. Upon navigation on clicking the “Get started for free” or “Join us today” it will navigate to the signup page.

5.1.2 Signup page:

This page is accessed by clicking on the 2 buttons and the “signup” in the top right of the navigation bar. This is very straight forward prompting the user to type in the necessary details, first name, last name, email address and password. This information is not grabbing very personal data as some others may do. Each section must be entered otherwise a popup will appear prompting the user as shown below.



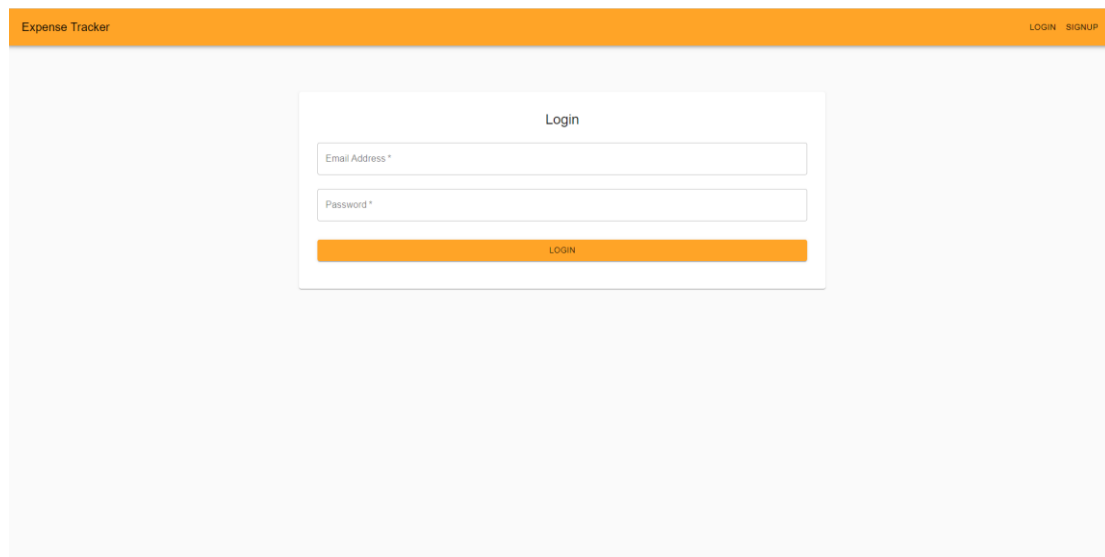
The image shows a form with two input fields. The first field is labeled "First Name *" and is empty. The second field is labeled "Last Name *" and contains the text "Last Name *". A red border highlights the "Last Name *" field, and a red error message box appears above it, stating "Please fill in this field.".

Once this information is submitted it is passed along to the MongoDB database through the post api using Axios as mentioned in 4.4.2. Once the password is passed along it uses Bcrypt to automatically encrypt the password. Bcrypt is an encryption tool used to encrypt the password making it inaccessible by the database shown below. This allows for security making the user feel safe.

```
password: "$2b$10$9w1SqnwNV6w/ry3VsU8PFcKs0Uzaehxkis.wz3MgwRH26IKyjbyK"
```

Once the user has signed up using the button to submit their data they will be redirected to the log in page.

5.1.3 Log in page:



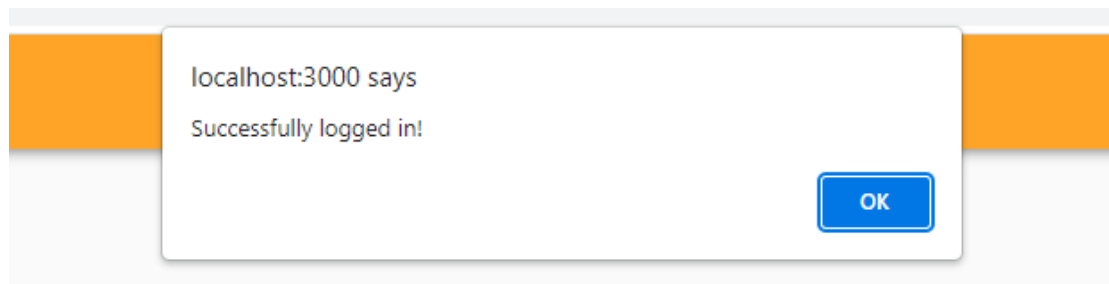
The image shows a login page for an application called "Expense Tracker". The page has a white background with a light gray border. At the top, there is a navigation bar with the text "Expense Tracker" on the left and "LOGIN SIGNUP" on the right. In the center of the page, there is a white box with a light gray border. Inside this box, the word "Login" is centered at the top. Below it, there are two input fields: "Email Address *" and "Password *". Below the input fields, there is a red button with the text "LOGIN" in white.

The login page is accessed by successfully signing up or clicking on the “log in” on the navigation bar. The user will then enter their details, if unsuccessful due to a wrong email or password it will display accordingly, as shown below.

Invalid email or password

This tells the user sticking to the user-friendly nature mentioned within my non-functional requirements.

When the user Logs in successfully by entering the correct information, the website will tell them using a popup, as shown below,

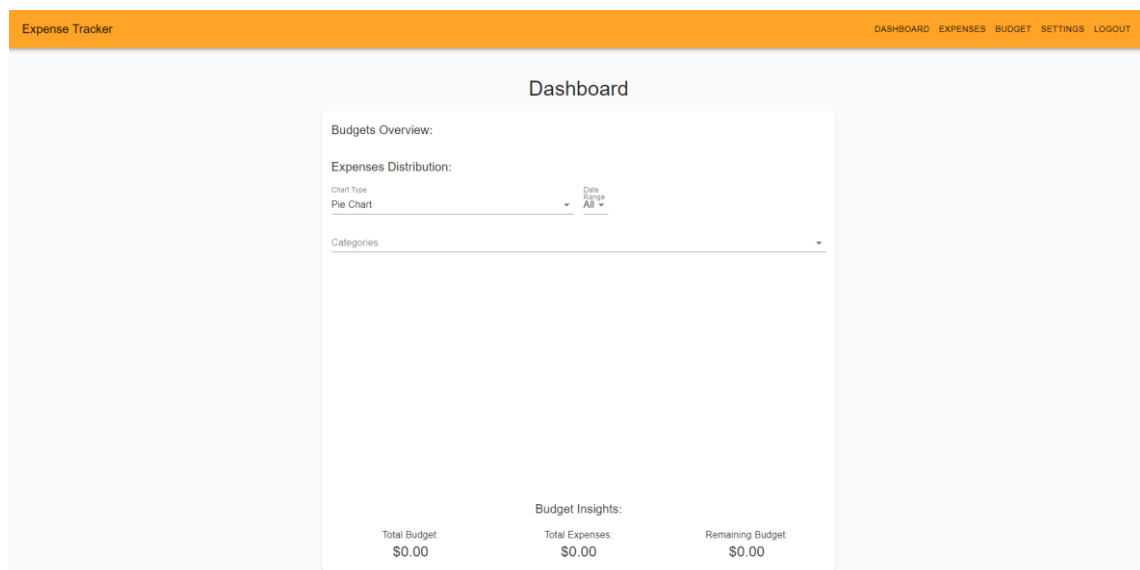


Login

Once the user logs in, they will be redirected to the main application passing the authentication stages.

5.1.4 Dashboard page:

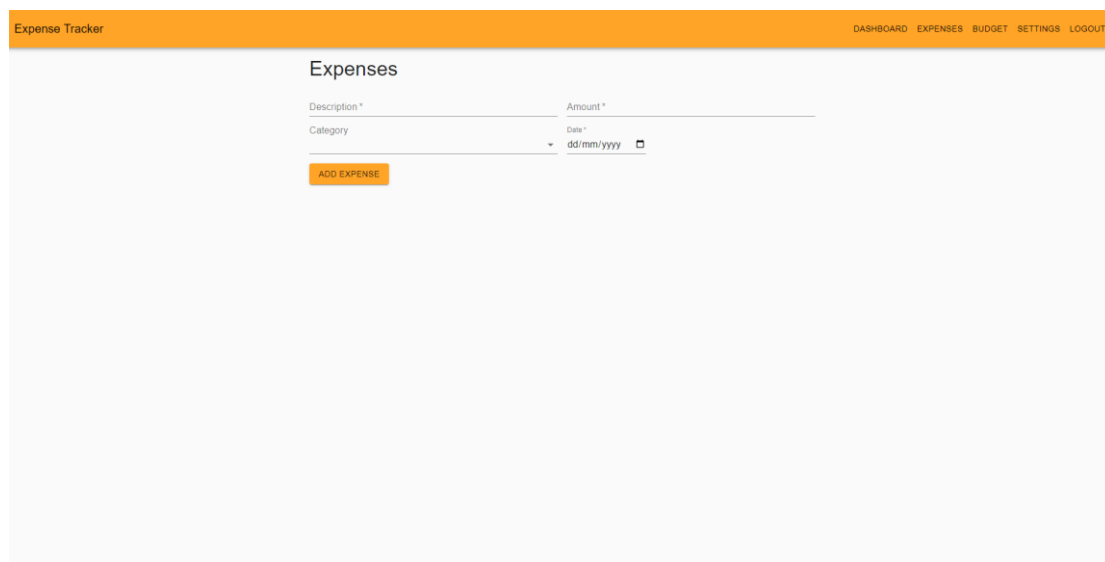
Once the user logs in successfully passing all of the authentication stages they will be greeted with the dashboard revealing the main application.



The dashboard is very clear to the user with a title in the middle of the page allowing the user to easily see what they are on. The theme of orange and black is shown throughout the site from the homepage to the current stage. At the moment the dashboard is empty due to the fact that no entries have been made within the expenses page and budgets page. However, it is very clear that this will be the case shown by the budget insights being 0, and no chart showing up when pie chart is selected. This section will be expanded on once the user has made entries in the necessary pages.

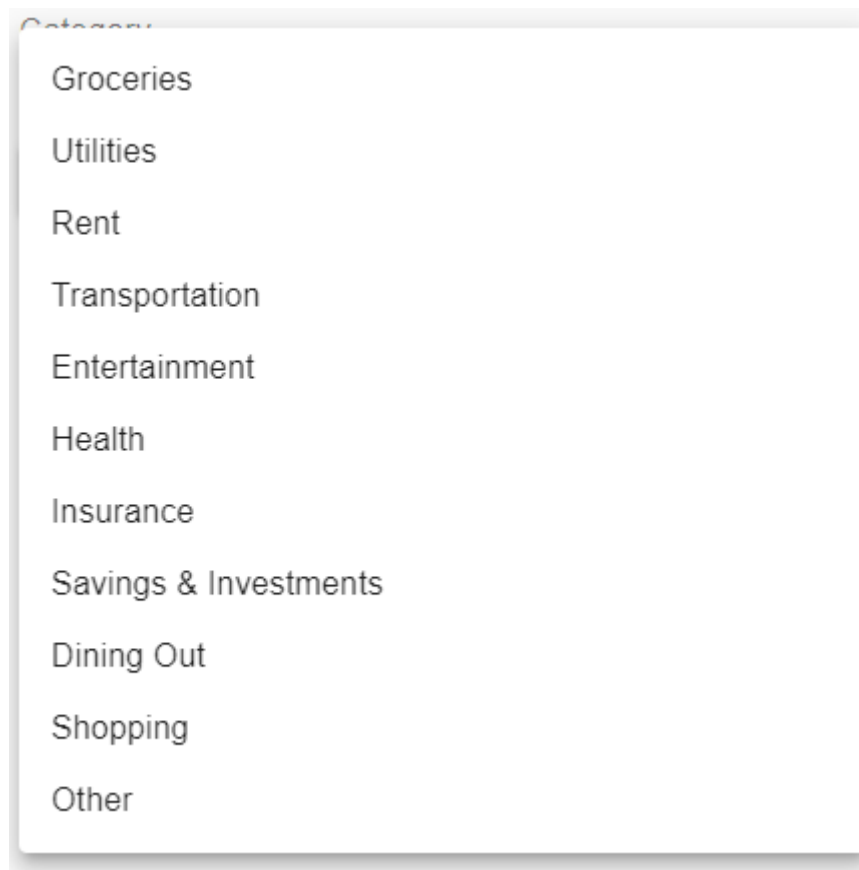
5.1.5 Expenses page:

The expenses page is one of the main within the application including the main functionalities to work with everything else. The page is shown below.



The screenshot shows the 'Expenses' page of an 'Expense Tracker' application. The page has an orange header bar with the application name on the left and navigation links (DASHBOARD, EXPENSES, BUDGET, SETTINGS, LOGOUT) on the right. The main content area is titled 'Expenses' and contains a form with the following fields: 'Description *' (text input), 'Amount *' (text input), 'Category' (dropdown menu), and 'Date *' (date picker set to 'dd/mm/yyyy'). Below these fields is an orange 'ADD EXPENSE' button.

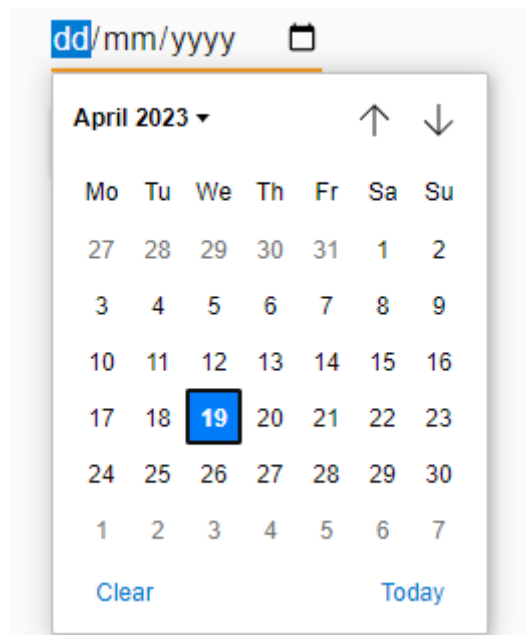
This looks relatively plain however it fills up once users add their expenses. The adding of the expenses is very straight forward and simple with prompts on what to include in each text box. The boxes consist of "Description", "Amount", "category" and the "date". The user will type in the description of their expense not too much just a small description will suffice. The amount which is how much their expense would be. And the category which is a dropdown menu allowing for many different options as shown below.



As you can see most of the normal expense categories are covered however if the user has one that is not on the list, they are able to select the other option. Upon choosing that another text box will appear allowing them to write in their own choice of category.

Description *	Amount *
Category	
Other ▼	Other Category *
Date *	
dd/mm/yyyy 📅	

This allows the user experience to be effectively used as they have no way of complaining that the website didn't include the right categories. Moreover, going back to the "date" once clicked onto a calendar opens up allowing for a visual view of the date allowing for easy use for the user.



This is very straight forward to use, also if they wish to click on today, it will display today's date.

Upon creation of the category, it will show up under the add expense button in a formatted inline table form. Below is what I will create.

Description *	Amount *
Test	200
Category	Other Category *
Other	category other test
Date *	
21/04/2023	
ADD EXPENSE	



When all the fields are filled in they can simply click expense resulting in:

Expenses

Description *	Amount *
Category	Date *
	dd/mm/yyyy
ADD EXPENSE	

Test

Amount: 200, Date: 21/04/2023, Category: category other test

As you can now see the data has been properly shown in a nice simple format easy to read and view. There are two buttons next to the expense, one is the edit button and the other is the delete button. Upon clicking the edit button, it will add the data into the boxes again and also change the "Add Expense" to "Update Expense" remaining straight forward for user use. This is shown below.

Expenses

Description *	Amount *
Test	200
Category	Date *
	dd/mm/yyyy

UPDATE EXPENSE

Test

Amount: 200, Date: 21/04/2023, Category: category other test

Now that this has been entered you are able to change the data according to your needs, in this example I will change “Test” to test 7 and the amount to 500, with the date to the 22nd. Upon clicking update, it will instantly show the changes.

Test 7

Amount: 500, Date: 22/04/2023, Category: category other test

The delete button just simply deletes the expense from the database.

Now that the user has created an expense its time to set a budget for that expense.

5.1.6 Budget Page:

This page is used for setting the budgets for the expenses that are created allowing for proper expense management sticking with the functionality requirements for this application. The view of this page is as shown below:

Expense Tracker

DASHBOARD EXPENSES BUDGET SETTINGS LOGOUT

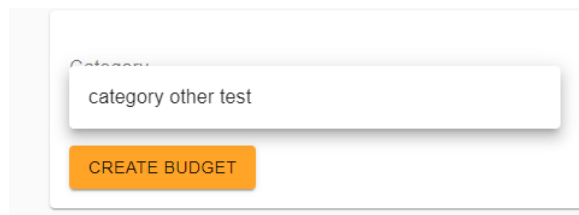
Budget

Category	Amount

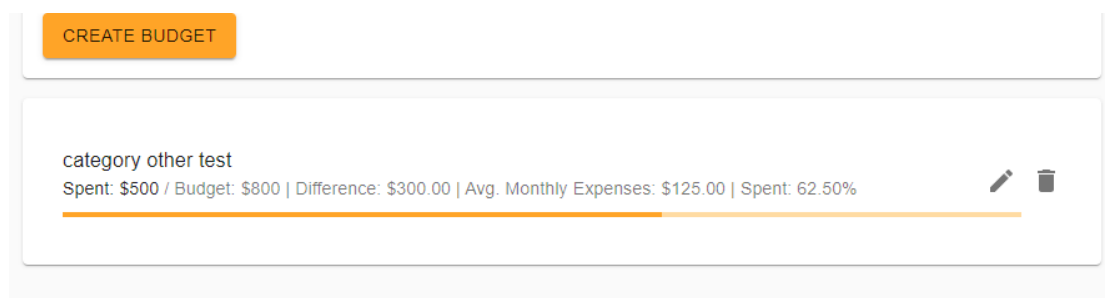
CREATE BUDGET

The simplicity of the design is consistent throughout the website as seen, furthermore staying with the approach of allowing the user to easily navigate and know what to do without any extra learning.

The “category” dropdown is the category for the expense that you have created, This will include all of the categories which are created within the expenses previously in 5.1.5. This dropdown will be shown below.



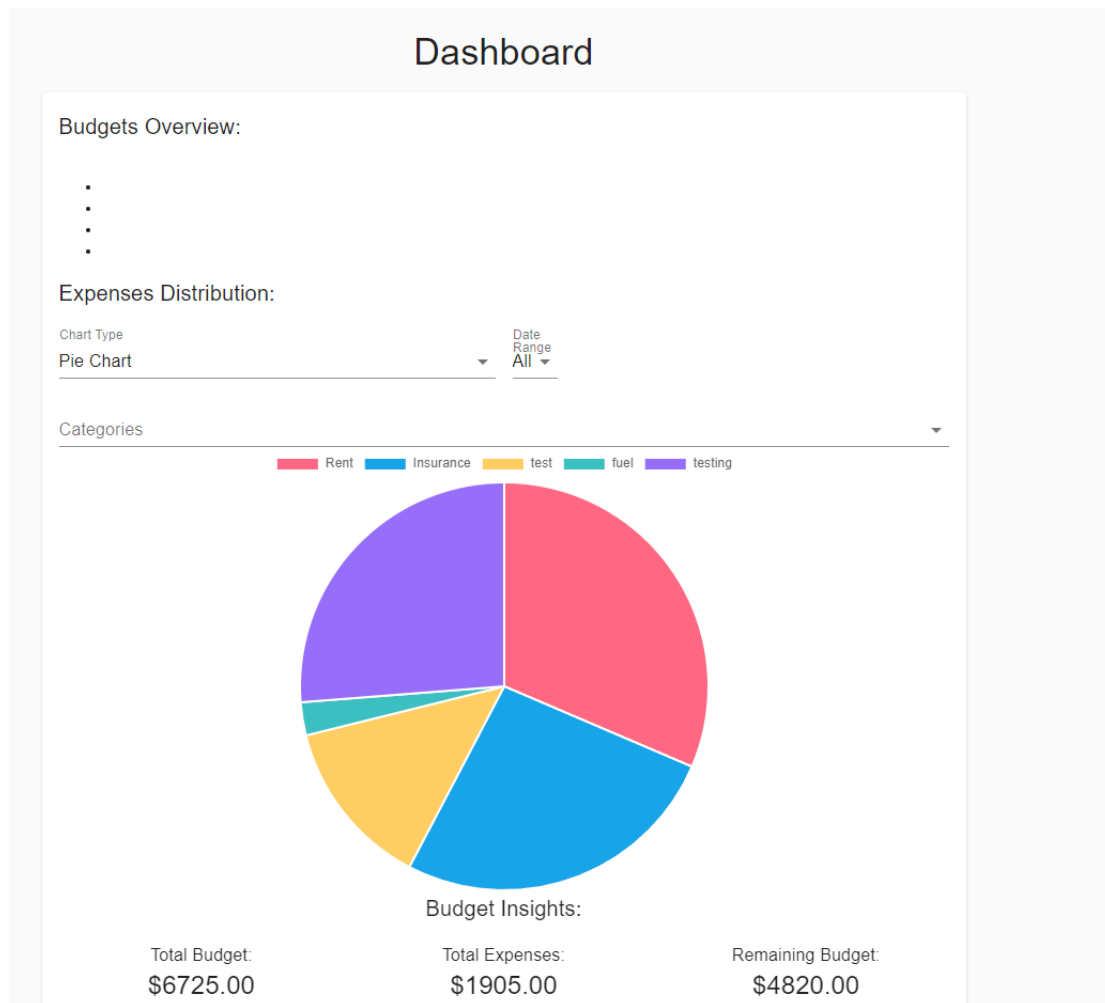
This is the category which was created in the previous section only shown. However, with more expenses and categories they will all show up. Upon Creation of the budget the user will need to enter the amount that they feel would be a suitable budget for their expense category limiting themselves in a structured way. Once this is created the budget will be shown below.



The budget is shown with the amount spent in the expenses also showing the budget amount next to it. The difference is simply the budget – expenses, which is shown next to it for more information for the user. Furthermore, it shows the calculation of the average monthly expenses which calculates it based on how much their average would be per month. Also showing the percentage spent of the budget. These analytics provide clear insights of their spending in a small format. Furthermore, the visual nature of the progress bar is helpful for some users as they can see how close they are getting to filling the bar. The user is able to edit their budget for the expense which was covered previously on how the editing would work, also the delete button will delete the budget in case they have a change of mind.

Dashboard Analytics:

The dashboard is the main feature of the application including the main view for analytics of the expenses, allowing for comparison between expenses which allows the user to make better financial choices. After creating their expenses and budgets of choice the user will see the pie chart shown below.



The user can see all their expenses listed in different colours, ensuring the view for different expenses and comparison. Additionally, the total expenses and budgets are shown at the bottom revealing to the user how their overall spending is. If the user wishes to change the chart view as this may not be ideal for everyone, they have a few different choices.

Chart Type

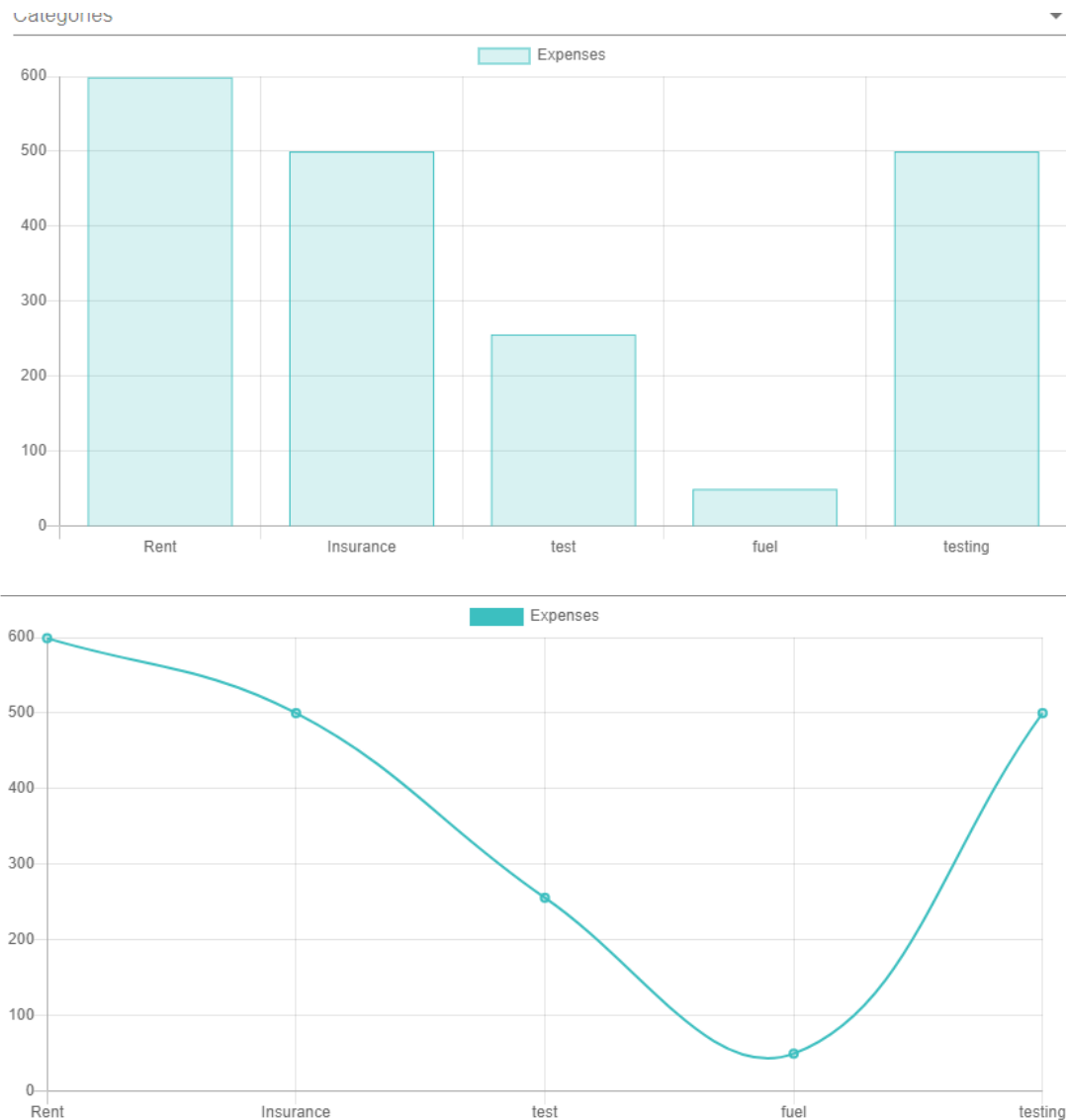
Pie Chart

Bar Chart

Stacked Bar Chart

Line Graph

This dropdown is revealed once the “Chart type” component is selected. There are a few different types of visual analysis choices for the user, allowing for a wider range adapting to the user’s differentiation in choices. The views for each chart will be shown below.



These are only 3 of the 4 charts as the stacked bar chart functionality is not working as intended and only displaying a normal bar chart. This can be fixed for future use by including the budgets directly next to the expense allowing for a clearer view. However, the main functionalities of pie chart, bar chart and line graphs work as intended shown with a clear view of their expenses allowing for visual interpretation for the user.

The user is also able to choose which expenses to show if they wish not to see them all at once. This will be done through the dropdown labelled categories displaying it below. The user can tick which expense categories they wish to see, and it will be shown accordingly within the chart.

☐ Rent
 ☐ Insurance
 ☐ test
 ☐ fuel
 ☐ testing

After the user is done analysing the chart and done with their monthly or weekly expense management, they can simply click the log out button at the top right of the navigation bar. Which will then return them to the homepage shown in 5.1.1.

5.2 Testing

In this section testing for the application will be carried out against the functional requirements listed in 4.2. Additionally, I will be a few tests against it and what results are given for each test also giving a short explanation of how it meets the functional requirements.

User Authentication tests:

Initial Test:	Expected Result	Actual Result	Met Requirement?
Sign up using email address and password with first name and last name	User successfully signs up	User successfully signs up	Yes, allowing the user to sign up
Sign up using a fake email	User gets rejected saying its not a real email	Allows user to sign up	No however there is no integration for this as it would require external api to solve issue.
Log in with correct details	Lets the user into the web application dashboard	Lets the user into the web application dashboard	Yes, as it allows proper sign in functionality.
Log in with incorrect email	Should come with an error saying that email is incorrect	Comes with an error saying email or password is incorrect.	Yes, as it produces proper protection for the application and data.
Log in with incorrect password	Come with an error saying	Comes with an error saying email	Yes, not allowing user in with incorrect

	password is incorrect.	or password is incorrect.	password using proper security.
Click log out	Log out the user returning to homepage	Log out the user returning to homepage	Yes, working log out function.
Refresh page	Keep the user logged in as they are still on the site	Logs out the user.	No as they should be logged out when they leave the site. Fixed by tracking user actions on the browser and knowing when they leave.

Budget Management:

Initial Test:	Expected Result	Actual Result	Met Requirement?
Creating a budget for category for rent	creates a new section underneath	creates a new section underneath	Yes, as it allows for creating a budget for a specific category
Create a budget for a category that's not created	will not work as you cannot budget something that doesn't exist	doesn't show up in the dropdown	Yes, as this is intended as a user needs to set create an expense category before budgeting.
After creating a budget, you can see it in the list below	it will appear after clicking create budget filling In the correct information.	Appears as it should in the list underneath	Yes, this is intended to show up and is correctly doing so.
Clicking edit on a budget	Fills in text box with information and changes create to update	Fills in text box with information and changes create to update	Yes, this works and applies to the functionality
Clicking delete on a budget	Deletes the budget from the list and database	Deletes the budget from the list and database	Yes, this works as it should and applies to the functionality.
Displaying correct information in the budget list insights	All calculations should appear as correctly	All calculations except average monthly expenses works as intended	Yes, however average monthly expenses can be fixed by properly defining the calculation and taking in the date ranges per month.
Creating a budget for an existing category	Shouldn't create it coming up with an error	Doesn't create it at all however doesn't tell the user either just disappears	Yes, works as intended however needs changing to tell the user that it's a duplicate for further user experience.

Expense management:

Initial Test:	Expected Result	Actual Result	Met Requirement?
Create an expense successfully including all information category, description, amount, and date.	Creates expense successfully adds it to the list below	Creates expense successfully adds it to the list below	Yes, creates expense successfully following the functional requirement
create expense only using numbers in description	Should say characters not allowed in description	it allows the expense to be created with the numbers in expense	No, as it wouldn't make sense for only numbers to be used in the description. Changes to check if numbers are present and if they are present a popup.
View list of expenses	Can see the list of expenses underneath creating them	Can see the list of expenses underneath creating them	Yes, works as intended following the functional requirement,
Edit an expense	Enters current information in text box and changes create to update	Enters current information in text box and changes create to update	Yes, this works as expected following the functional requirement.
Delete an expense	Deletes expense from the list and database	Deletes expense from the list and database	Yes, this works following the functional requirement.
Filter expenses by category	Allow to tick and untick categories wanted and display accordingly	Can tick and untick categories wanted and displays it accordingly	Yes, this works as expected and follows the functional requirement.
Filter expenses by this week in date range	Shows all the expenses this week in the chart	Doesn't show any expenses in the chart for this week	Doesn't work as intended can be fixed with proper checking of the date's expenses are created and comparing with today's date.
Filter expenses by this month	Show all expenses within the current month	Shows all expenses in the current month	Yes, this function works as intended following the functional requirements.
Filter expenses by last month	Show the expenses from the previous month	Shows the previous months expenses.	Yes, this function works as intended showing last months expenses in

			the chart following the functional requirements.
Filter expenses by this year	Shows the expenses within this year	Doesn't show anything at all	No, this function doesn't work as intended. Doesn't follow the functional requirements.

Data Visualisation:

Initial Test:	Expected Result	Actual Result	Met Requirement?
Tick 3 random expense categories and show accordingly in the pie chart	Shows the 3 expenses correctly in the pie chart	Shows the 3 expenses correctly in the pie chart	Yes, this successfully meets the functional requirement of choosing category of choice
Click the drop down and choose the pie chart	Shows pie chart with all current expense categories	Shows pie chart with all current expense categories	Yes, this meets the function of pie chart included
Click the drop down and choose the bar chart	Shows bar chart with all current expense categories	Shows bar chart with all current expense categories	Yes, this shows the bar chart successfully meeting the requirement
Click the drop down and choose the stacked bar chart	Shows stacked bar chart with all current expense categories	Shows stacked bar chart with all current expense categories	No, this doesn't show a stacked bar chart not meeting the requirement, a fix for this can be to add the budget for the expense directly next to it in the chart
Click the drop down and choose the line graph	Shows the line graph with all expense categories	Shows the line graph with all expense categories	Yes, this works following the functional requirement, and working as intended.

Overall, throughout testing against the functional requirements I have found some issues. These consist of the expense date filters, not catering for "this year" and "this week" filter. These can be fixed by making sure it collects all the dates from the relevant expense categories and displaying it correctly within the filter choice. There

may be some errors within the code for these 2 functions. Additionally, another problem I had found was the stacked bar chart not working as intended. This is due to the budget data not being pulled through and displayed next to the right category. This can be fixed by displaying the budgets next to the category in the stacked bar chart in the functions.

6 Project Management

6.1 Professional Issues

Legal and security:

There are limited legal issues that may arise when developing this web application.

The Data Protection Act 2018 must be considered as users' personal data, such as emails and passwords, will be stored [21]. However, this will be mitigated as the data is stored in a secure database which will be inaccessible to external users.

When passwords are used, it is automatically encrypted to ensure further privacy.

Compliance with the UK Copyright, Design & Patents Act 1988 is required [22]. As such, the intellectual property of this dissertation and project will be owned by Oxford Brookes University.

Ethical:

An ethical consideration was the notion that some users may have accessibility issues. As Web Content Accessibility Guidelines highlights, there is a need to acknowledge users with impairments such as mobility or cognitive difficulties [23]. The colour contrasts allow for clear text and differentiable tabs. There are also visual aids included such as graphs to allow for a better understanding of the user's expenses. Therefore, this enables anyone to use this application without feeling concern.

Social:

Certain social issues are tackled with the creation of this web application. For instance, there is an assumption that expense trackers should only be utilized by professional users. However, this application takes a different approach, which is easy to use and straightforward. The design of the application is simple to navigate and has been streamlined to only contain necessary information. In turn, this promotes efficiency and a likelihood of a wider audience adopting this application.

6.2 Risk Assessment and Mitigation

Potential Risk	Potential Causes	impact	Likelihood	Risk	Mitigation
Loss of work	Poor usage of Git version control	4	3	12	Make sure I use branches off the master and comment updates effectively on commit
	Loss or damage of personal computer	5	2	10	Ensure all work is backed up to the cloud
Learning Curves	Taking too long in learning	5	2	10	Stick to gantt chart and make sure all learning is structured in a way to learn it all also practice implementation
Struggles with code	features	3	5	10	Try using community approaches, debugging, and looking for material online. Consult with others if unsolvable.
Missing deadlines	Illness	1	3	3	Follow doctors advice and apply for exceptional circumstances. Also ensure that my supervisor is aware.
	Mismanagement of timing	4	3	12	Make and follow the gantt chart. Also ensure other aspects of life outside the project are managed correctly.

6.3 Project Timeline

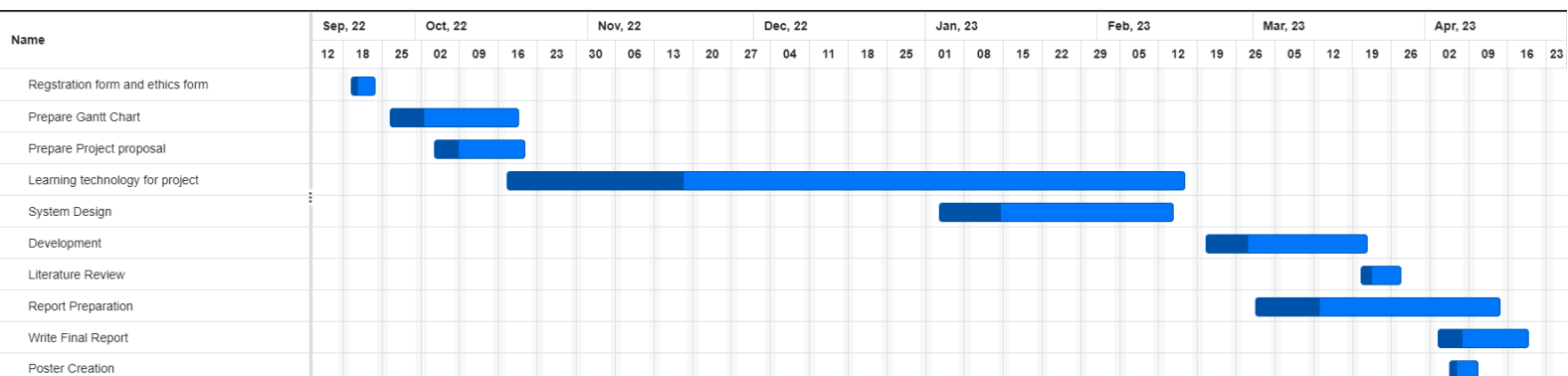


Figure: Gantt Chart

Many struggles occurred when following this gantt chart as situations had taken place in life not making me follow this thoroughly however, following it the best as possible, I had managed to change a few things around mentally. Getting through the project while following the gantt chart was very helpful however sticking to it was the tough part which wasn't done well.

7 Conclusion

This project focused on the development and approaches of the creation of an expense tracking application. Throughout the process many different technical approaches were observed and implemented. These have been compared in the methodology section stating my reasons for picking the technology chosen within the project. Also addressing a different approach within the Literature review helped thoroughly when deciding. The methods used can be interpreted to show what the development would look like and the results which are shown in the project allow for a better understanding of what the application achieved.

There is much room for improvement in this project, mainly due to time constraints faced and effective time management was a huge issue for me. Furthermore, I would've loved to do the testing straight after the development was completed allowing me more time to correct the issues that had been found.

7.1 Future plans

Direction for a future study:

One limitation of the research on the expense tracking applications was that there was a general lack of literature found. This was due to the data gathered being outdated and didn't reflect modern technology. Hence, this prevents the ability to fully understand the benefits and current access to expense tracking. Further research is vital to provide an analysis on the advancements of expense tracking technology in financial management.

7.2 Self reflection

One skill I have learnt during the development process is problem solving. This was demonstrated by overcoming difficulties presented with researching and programming. Through this I have gained a thorough understanding of technology used such as React, Express.js, and MongoDB. Another skill I have taken away from this project is an improvement in my adaptability. Initially, I neglected help from my dissertation supervisor but then began to converse and discuss feedback given. This ultimately led to a smooth writing process and unique insight on my project.

One aspect that I wish to improve is my time management skills. It was challenging to balance my leisure time and the obligation to complete my project. In turn, this resulted in decreased productivity and failure to account for unexpected events. I have learnt that it is important to set priorities whilst maintaining a healthy wellbeing to avoid a lower quality of life. Upon completing this project, I have succeeded in developing a clear and effective application that meets the specified requirements. As such, I demonstrated the ability to learn new technologies and apply it to the project.

Bibliography:

- 1: Acevedo, S. and He, E. (2023). *What Are the Best Budgeting Apps of April 2023?* [online] Business Insider. Available at: <https://www.businessinsider.com/personal-finance/best-budgeting-apps?r=US&IR=T#the-best-budgeting-apps> [Accessed 20 Mar. 2023].
- 2: My Money Coach. (2019). *Did You Know That an Expense Tracker Can Solve All of Your Budget Planning Problems?* [online] Available at: <https://www.mymoneycoach.ca/blog/expense-tracker-solve-budget-planning-problems.html> [Accessed 20 Mar. 2023].
- 3: The Interaction Design Foundation. (2016). *What is Human-Computer Interaction (HCI)?* [online] Available at: <https://www.interaction-design.org/literature/topics/human-computer-interaction> [Accessed 22 Mar. 2023].
- 4: Latest Technology Insights, Web & Mobile Blog | Arka Softwares -. (2022). *A Complete Guide To Develop Efficient Expense Tracking App.* [online] Available at: <https://www.arkasoftwares.com/blog/expense-tracking-app-development/> [Accessed 26 Mar. 2023].
- 5: Singh, S. (2020). *What is Testing in Software? — The Three Main Types of Testing Explained in Simple English.* [online] Medium. Available at:

<https://levelup.gitconnected.com/what-is-testing-in-software-the-three-main-types-of-testing-explained-in-simple-english-da0fec7ae5d6> [Accessed 20 Mar. 2023].

6: Pulse Headlines. (2018). *4 Target Audiences for Your Finance App*. [online] Available at: <https://www.pulseheadlines.com/4-target-audiences-finance-app/72045/> [Accessed 26 Mar. 2023].

7: mint.intuit.com. (n.d.). *Budget Tracker & Planner | Free Online Money Management | Mint*. [online] Available at: <https://mint.intuit.com/>. [Accessed 26 Mar. 2023].

8: Payne, K. (2022). Mint Budgeting App Review. *Forbes*. [online] 7 Mar. Available at: <https://www.forbes.com/advisor/banking/mint-budgeting-app-review/> [Accessed 10 Apr. 2023].

9: YNAB. (2021). *YNAB. Personal Budgeting Software for Windows, Mac, iOS and Android*. [online] Available at: <https://www.ynab.com/> [Accessed 10 Apr. 2023].

10: Cox, D. (2021). *YNAB Review: 5 Things To Know About the Budgeting Tool*. [online] Clark Howard. Available at: <https://clark.com/save-money/ynab/> [Accessed 10 Apr. 2023].

11: Wally 3.0. (2015). *Home | Wally 3.0*. [online] Available at: <https://www.wally.me/> [Accessed 10 Apr. 2023].

12: Harad, M. (2022). *Wally App Review - Strong Budgeting App Relying On In-App Purchases*. [online] Overdraft Apps. Available at: <https://overdraftapps.com/wally-app-review/> [Accessed 10 Apr. 2023].

13: ThemeSelection (2022). *Best Web Development Stack 2023*. [online] DEV Community. Available at: https://dev.to/theme_selection/best-web-development-stack-2jpe [Accessed 10 Apr. 2023].

14: Aggarwal, S. (2018). Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects*, [online] 5, pp.133–137. Available at: <http://ijrra.net/Vol5issue1/IJRRRA-05-01-27.pdf>.

15: Turcomat.org. (2023). *View of Spending Tracker: A Smart Approach to Track Daily Expense*. [online] Available at: <https://turcomat.org/index.php/turkbilmat/article/view/8759/6829> [Accessed 12 Apr. 2023].

- 16: MindManager Blog (2021). *Your introductory guide to the waterfall methodology | MindManager Blog*. [online] MindManager Blog. Available at: <https://blog.mindmanager.com/waterfall-methodology/> [Accessed 15 Apr. 2023].
- 17: Sommerville, I., Columbus, B., New, I., San, Y., Upper, F., River, S., Cape, A., Dubai, T., Madrid, L., Munich, M., Montreal, P., Delhi, T., São, M., Sydney, P., Kong, H., Singapore, S. and Tokyo, T. (2011). *SOFTWARE ENGINEERING Ninth Edition*. [online] Available at: <https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf>. [Accessed 15 Apr. 2023].
- 18: Pressman, R.S., Ince, D. & Lowe, D. (2000). *Software Engineering: A Practitioner's Approach (European Adaptation)*. McGraw-Hill Education. [Accessed 15 Apr. 2023].
- 19: Reactjs.org. (2021). *JSX In Depth – React*. [online] Available at: <https://legacy.reactjs.org/docs/jsx-in-depth.html> [Accessed 15 Apr. 2023].
- 20: Shah, H. (2022). *7 Frontend JavaScript Frameworks Loved by Developers in 2022*. [online] Simform - Product Engineering Company. Available at: <https://www.simform.com/blog/javascript-frontend-frameworks/> [Accessed 17 Apr. 2023].
- 21: Government Digital Service (2011). *Data protection*. [online] GOV.UK. Available at: <https://www.gov.uk/data-protection> [Accessed 17 Apr. 2023].
- 22: Intellectual Property Office (2011). *Copyright Act*. [online] GOV.UK. Available at: <https://www.gov.uk/government/publications/copyright-acts-and-related-laws> [Accessed 17 Apr. 2023].
- 23: Wwww.gov.uk. (2022). *Understanding WCAG 2.1 - Service Manual - GOV.UK*. [online] Available at: <https://www.gov.uk/service-manual/helping-people-to-use-your-service/understanding-wcag> [Accessed 17 Apr. 2023].

Appendices:

Link to poster:

https://drive.google.com/drive/folders/1S0ynUiBpc8VD9mj0AQHhXe55DoJOUfR-?usp=share_link

Link to video recording:

https://drive.google.com/drive/folders/1NIVQccNDiNUo5P0_aFnjNQgwXASvvkwl?usp=share_link

Link to source code:

https://drive.google.com/drive/folders/1wMbKK5lcwMsDLLr6lx5-3HEc6DnZJWo1?usp=share_link