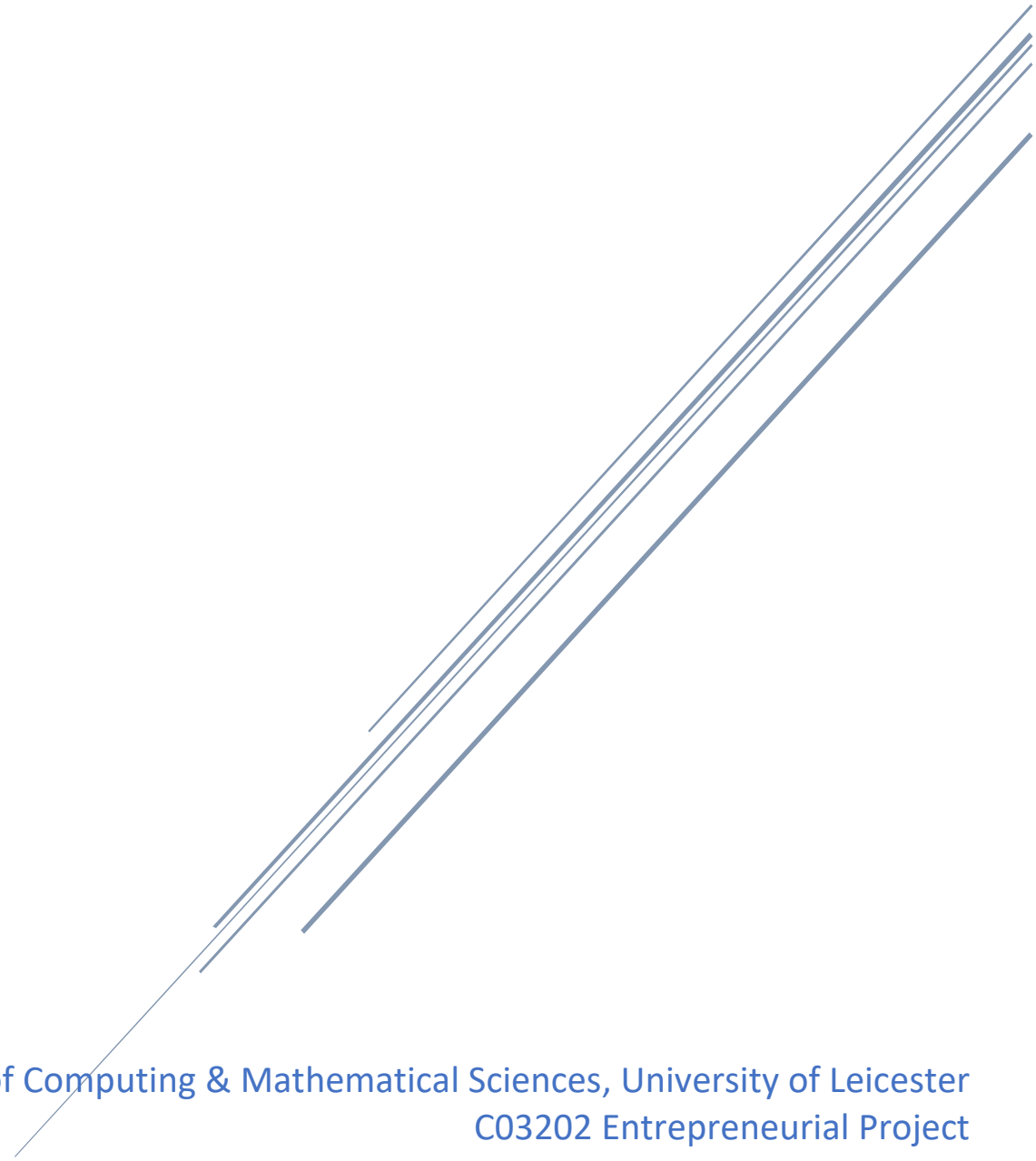
DISSERTATION

Software for the fundamentals of accountancy

Kazim Manji



School of Computing & Mathematical Sciences, University of Leicester

C03202 Entrepreneurial Project

May 2023

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s).
Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s).
I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name: Kazim Manji

Signed:

Date: 10/04/2023

Contents

Abstract 3
Introduction 3
Survey Of Literature/Information Sources 4
Business Development 4
Unique Value Proposition 4
Lean Canvas 5
The Customers 5
The Market 5
Market 5
Ways to market 6
Competition 7
Strategic Partnerships 11
Revenue Model 12
Monetisation 12
Revenues & Expenses 12
Design & Implementation 14
Requirements 14
Overview 14
Frontend 14
Backend 15
Security 15
Database 15
Schema 15
Results & Discussion 17
User Interface 17
Project Structure 23
Frontend 23
Backend 25
Testing 26
Critical Appraisal 27
Summary of work done & Critical analysis of work done: 27
Discussion on Social, sustainable, commercial, economic context 28
Personal development 28
Conclusion 28
References 29

Abstract

This report is an analysis of the work performed by myself while developing a bookkeeping application for my final year project with a focus on business development.

The business development work was carried out included creating a unique value proposition to clearly describe the benefits of my application to the customer. Analysis on the competition and creating lean canvases to better understanding their business and unique selling points and what specific customers they cater to. Identifying my own customer segment and understanding their needs and pain points to improve my software design and marketing was conducted as well as market analysis. Suitable marketing methods were recognized such as advertising platforms, social media campaigns, and content marketing strategies to name a few. A revenue model has been created and estimates of revenues and expenses have also been described.

The web application has been built and the software design and implementation has been described including the technologies used, security of the application and database design. Finally, the results have been analysed followed by a critical appraisal.

Overall, the application is functional and meets the challenges set at the start of the project. However, there are more features that could be implemented which would make the application more competitive to attract a larger customer segment such as payroll capability which caters to medium sized businesses and other small enhancements such as automation and UI improvements.

Introduction

My bookkeeping application is targeted at small business owners or sole traders with a fundamental understanding of accountancy. Users who don't need many of the features that clutter the UI and confuse the customer like that which is offered by my competition such as Xero, FreshBooks and QuickBooks. The application is a practical and straight forward alternative that gives the user full control over which ledgers they wish to make double entries into. This means they can create their own accounts in a transparent manner by being able to directly see the T-accounts that they are making entries into as well as edit or delete those entries. The necessary financial reports such as the Profit and Loss statement and the Balance sheet are generated automatically and overview reports such as bar and pie charts help to give a quick financial overview of the business. The application gives what is needed to the customer is a clear manner compared to the competition where users may need training to operate it [19].

What makes this project unique is its simplicity and transparency as users may edit and delete records and are also not overwhelmed by large numbers of features. Users can view ledgers in the T-account format that they are taught when learning accounting which is completely familiar to them to avoid confusion. When making journal entries they also have full control over which accounts they can make entries into.

My aim is to build a web application to support the basics of accountancy. The application will allow bookkeepers for small businesses to input and track their business transactions for accounting purposes. For example, their income, expenses, assets, liabilities. The application allows them to get

an idea of where their business stands financially during their financial year by generating financial statements, namely the statement of financial position & the statement of profit and loss.

To complete the development, these objectives must be met:

1) Study existing systems to understand their implementation and improve upon what they have built.
2) Design a user interface which makes it easy to input financial data.
3) Design a database system that correctly stores data inputted.
4) Build both the frontend and backend of the application.
5) Build functionality to produce financial reports.
6) Test application on sample accounting data.

Survey Of Literature/Information Sources

To better understand how to build my application I decided to look at other accounting applications currently available. I looked at their user interfaces to see how other applications display their data. I wanted to understand how their interfaces reflected the type of customers that they cater to. This helped me when designing my implementation as I would then know what my audience is used to seeing. I also wanted to see the types of features they offered and if I could use or develop those features. This research can be found in the business development chapter.

I also researched my competitors to understand their unique selling points and created lean canvases for them to better understand their business. By understanding my competitor's business, I was able to improve the understanding of my own and what my strengths and weaknesses were. I was able to understand my main costs, resources used and potential partners. This is also found in the business development chapter.

I also had to conduct research into different software frameworks and libraries in the design and implementation chapter. I researched libraries such as Next.js, React.js and PostgreSQL to better understand which technologies would suit my applications needs. I investigated different backend frameworks such as Django which uses python or Express.js which uses JavaScript and was able to weigh up the positives and negatives behind each of them before choosing one [10].

My application also needed security which meant I had to understand how I could authenticate users using JSON web tokens as well as different security techniques [3]. I can protect my database from SQL injection attacks by utilizing an object relational mapper [11].

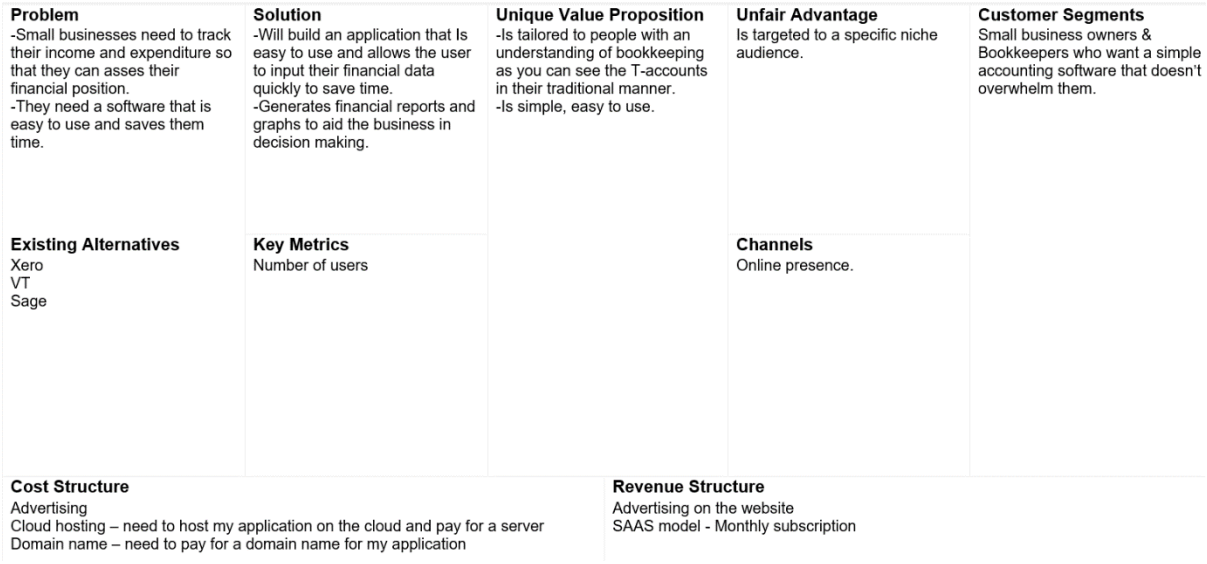
Business Development

Unique Value Proposition

"Fundamentals of accountancy"

A, simple to use, web-based bookkeeping application that allows the user to bookkeep using traditional T-accounts and gives the user full control over their accounts. They can quickly view their financial standing through reports namely the profit & loss account and the balance sheet.

Lean Canvas



Problem	Solution	Unique Value Proposition	Unfair Advantage	Customer Segments
-Small businesses need to track their income and expenditure so that they can asses their financial position. -They need a software that is easy to use and saves them time.	-Will build an application that Is easy to use and allows the user to input their financial data quickly to save time. -Generates financial reports and graphs to aid the business in decision making.	-Is tailored to people with an understanding of bookkeeping as you can see the T-accounts in their traditional manner. -Is simple, easy to use.	Is targeted to a specific niche audience.	Small business owners & Bookkeepers who want a simple accounting software that doesn't overwhelm them.
Existing Alternatives Xero VT Sage	**Key Metrics** Number of users		**Channels** Online presence.	
Cost Structure Advertising Cloud hosting – need to host my application on the cloud and pay for a server Domain name – need to pay for a domain name for my application			**Revenue Structure** Advertising on the website SAAS model - Monthly subscription	

The Customers

The target audience is for business owners with an understanding of the fundamentals of accountancy and want full control over their accounts. People who don't want to get overwhelmed by the many advanced features that other bookkeeping services provide and just want a simple and easy to use bookkeeping application. Currently my application is specialized towards small businesses and sole traders with an understanding of accountancy as it requires users to manually make double entries into the correct ledgers. Since there are no features for payroll, invoicing, payments, my application focuses on small businesses that don’t need this feature.

As I add more features, I could end up adjusting my customer base for example if I were to create a payroll feature it would be more attractive to customers with medium sized businesses who have many employees. I do plan on releasing more features in the future to encompass more business types and attract different customers but for now my focus is on small businesses since I don’t have the features that could compete with the competition in that target customer segment.

In the future I plan to include more automation to my application to attract customers that don’t have as much knowledge of accountancy and who are the primary target of my competitor such as FreshBooks. I could also keep the option for users to choose whether they would want automation or not allowing me to please both sets of customers.

The Market

Market

My market would be those who need a bookkeeping application which would be normally accountants or sole traders. The number of sole traders in the UK is 3.1 million [1]. Each of these individuals would need to carry out bookkeeping tasks which gives us a fairly large number of customers to target. The market size for accounting and auditing is $5.6 billion in the UK [2]. This means there is a good chance to gain customers as the market is quite large.

Ways to market

The way I plan to expand in this market is to focus on my UVP and expand my product to have more useful features that have not been implemented yet. If I can make sure my product solves the customers pain points and is effortless to use, then I can attract more customers away from my competition seamlessly. I could also copy some of the unique features that my competition has such as being able to connect to a user's bank account and automatically record and journal their transactions in their corresponding ledgers. This would make my application easier to use and more marketable. If I can focus on making bookkeeping simple and effortless for the user and allow them to bookkeep in a frictionless manner It will make my product more attractive to my customers.

There are many different methods to advertise my product such as posting my application on websites such as producthunt.com or Indiehackers.com. These websites allow you to freely share your new business and interact with potential customers as users can upvote your product and comment on it.

I also plan to advertise on social media websites such as on Facebook, Twitter, Instagram and reddit by creating social media pages for my business. By engaging with the communities on these sites by commenting and liking posts I could drive traffic to my website.

I can also use content-based marketing strategies to produce accounting-based content that my target customers, such as sole traders and small business owners, my find useful. I could create content such as videos, podcasts, articles or by writing a blog for either my own website or by posting on another website as a guest. This will allow my business will reach my target customer through the content I produce, and I would link my product and showcase my unique value proposition to drive traffic to my website and application. I also plan to upload my content to communities on social media platforms that are specific to my industry to further increase my outreach.

I also plan to also advertise through email marketing by building an email list with potential customers and emailing promotional content to those subscribed. If users subscribe to my email list, then they may receive promotional discounts as an incentive.

Customers could also find my business through their search engine which is why I used Next.js and server-side rendering to improve my search engine optimization. I also will implement other strategies to improve SEO such as keyword targeting and including meta data in my website [9].

I plan to use google search advertising as it more suitable for my needs and works well for business-to-business services [8]. Google adverts tend to get in front of people actively searching for a solution to a current need [8]. I want to provide an alternative bookkeeping application to those who are searching for one. Google ads will match my advertisement with search queries based on keywords such as accounting software, bookkeeping and even the names of my competitors. When users search Google for my competitors or any accounting software, my advertisement will also show up as a search result and this way I advertise to people who are more likely to need my product.

These adverts provide analytics so I can see how many impressions my advert has had and how many people were referred to my website allowing me to test different advertising methods and choose the most effective one in the future.

Competition

Direct Competitors

Figure 1: Business Model Canvas for direct competitors

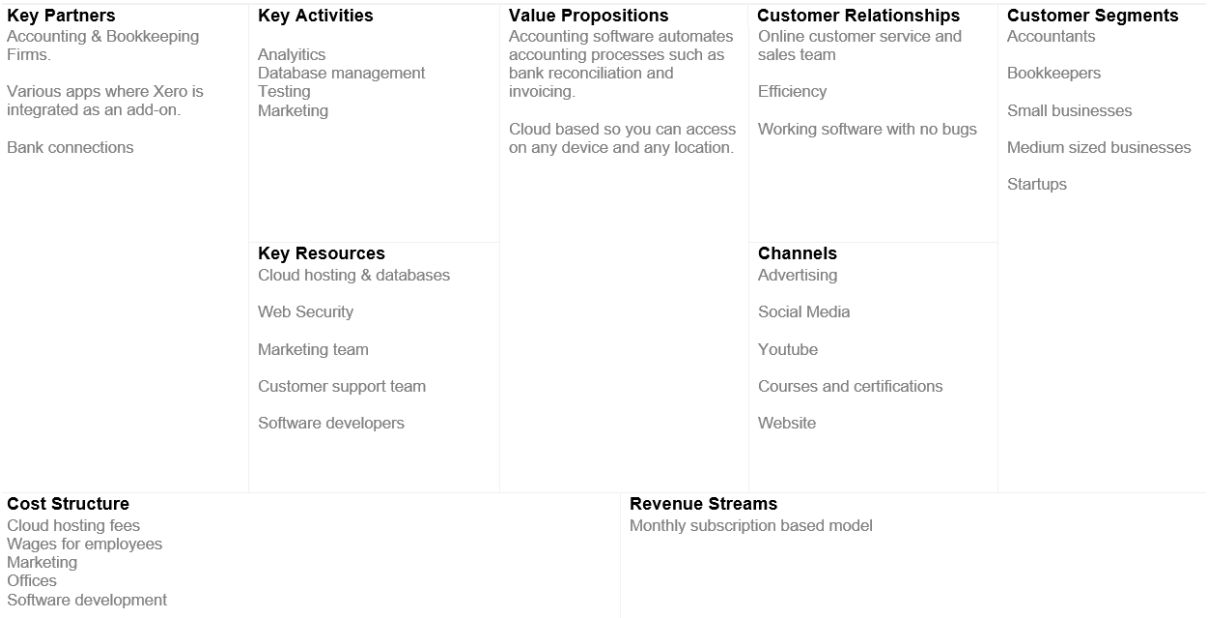


Figure 1 shows the lean canvas for all my direct competitors. I have collapsed each competitor into the same lean canvas because they are all very similar.

1) Xero [4]

Xero is a cloud accounting software for small businesses [4]. Xero provides many services on top of bookkeeping such as Payroll and VAT submission which makes it quite competitive for medium sized businesses. My application will only focus on the bookkeeping aspect for now. A competitive feature of Xero is that it connects directly to your bank account which saves the user time since they don't have to manually enter any bank transactions. This feature improves bookkeeping for the user and is something that I wish to implement in the future.

Strengths

- Automation of bank transactions, invoices, payroll.
- Specialized for medium sized business with accounting teams.
- Cloud based can access anywhere.
- Documents can be stored along with its ledger transaction so no need to keep paper copies.
- Mobile app to view data on the go.
- Video tutorials.

Weaknesses

- Can be quite expensive at £28 a month [4].
- Lots of features means a steeper learning curve.

- Difficult UI

Figure 2: Xero's reporting dashboard

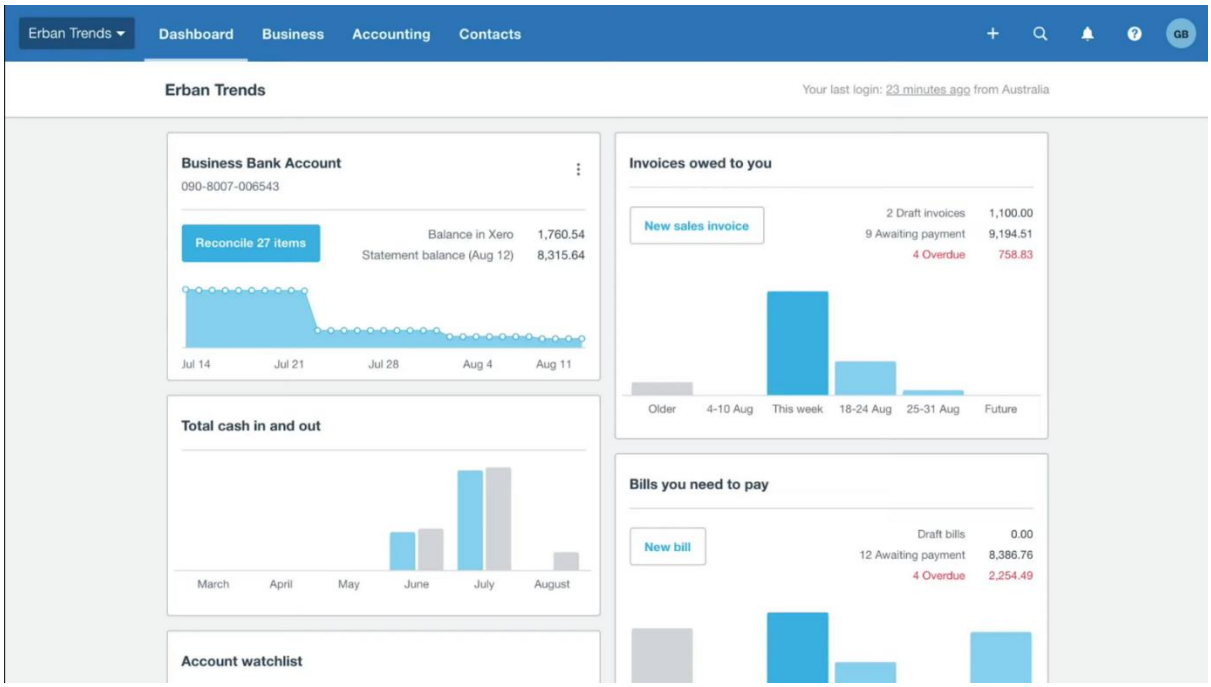


Figure 3: Xero's expense input page

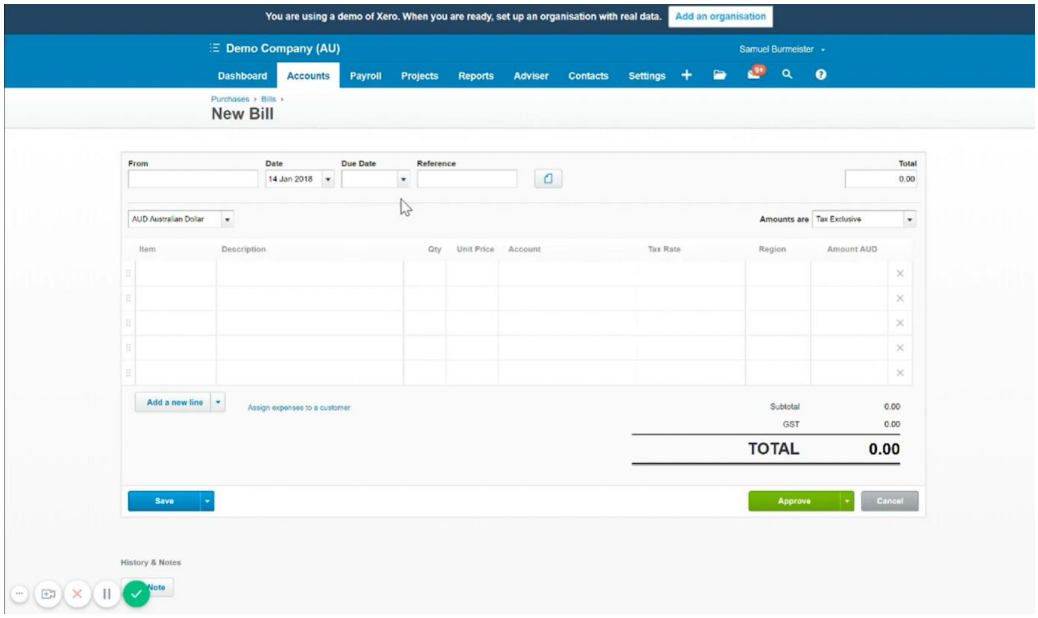


Figure 2 shows how Xero can give reports to the user using graphs and charts. These reports are very meaningful and is something that I believed I could implement to enhance my application. Figure 3 shows how Xero uses invoices to make entries into its expense account. My application would let the user manually make journal entries instead.

1) FreshBooks [5]

FreshBooks

Strengths:

- Specialized for freelancers.
- Project management features such as time tracking.
- Intuitive UI,
- Cloud based can access anywhere.,
- More affordable.

Weaknesses:

- Lacks comprehensive reporting.
- Lack of live customer support.
- No workflow integrations

Figure 4: FreshBooks time tracker [5]

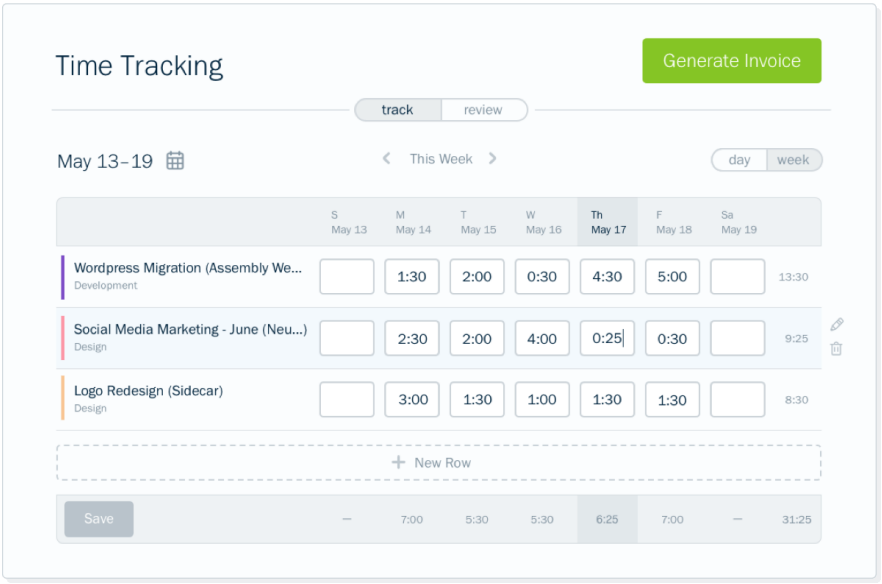


Figure 4 shows the time tracking feature to allow freelancers to record how many hours they have worked on a project. FreshBooks will then process invoices for them which saves the freelancer time as they don't have to input any entries into ledgers but just record their time.

2) Intuit Quick Books [13]

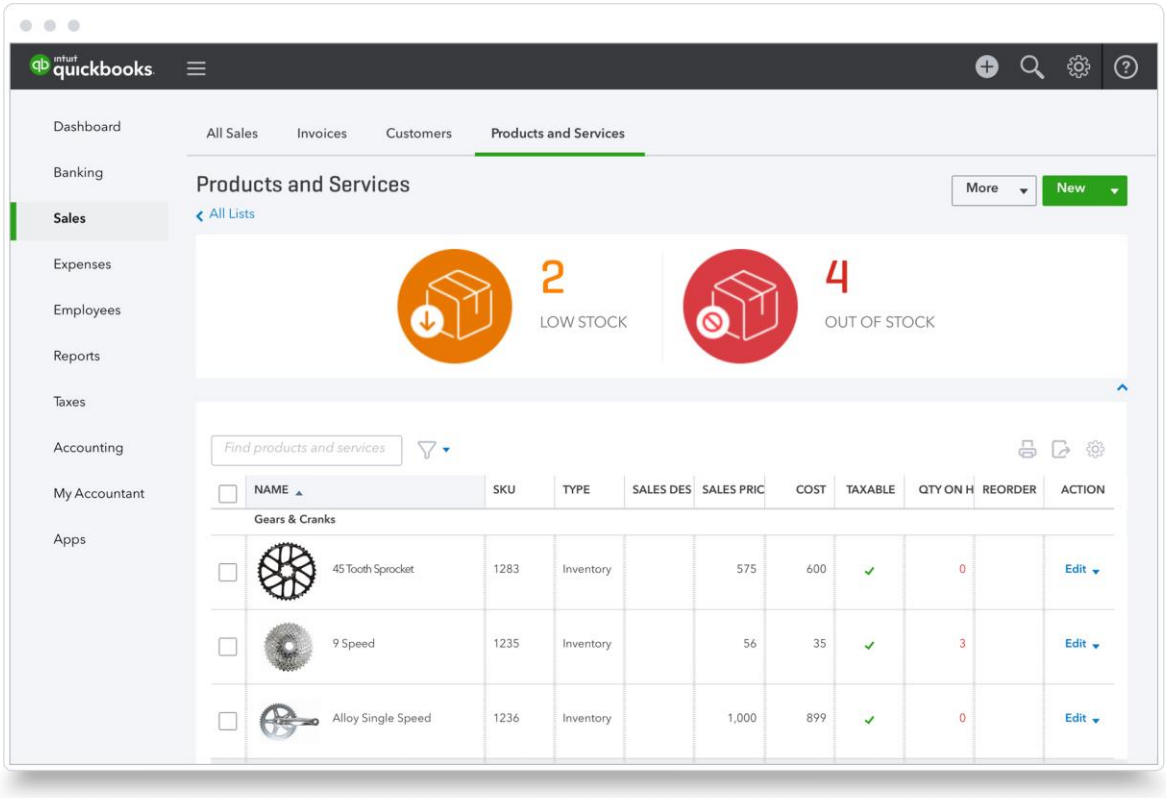
Strengths:

- Specialized for retail stores and small business without an accounting team.
- Scalable software with many different features.
- Affordable pricing.

Weaknesses:

- Lacks reporting.
- Limited number of users per company.

Figure 5: QuickBooks Inventory management system [13]



QuickBooks is good for businesses that sell products as you can sync your business transactions from other services such as PayPal, Square and Shopify to save the user time [20]. It also has an inventory management system as seen in figure 5. QuickBooks is more suitable for medium sized businesses due to these features.

Indirect Competitors

Microsoft Excel

Excel has the functionality to be used to track financial data. The issue with excel is that the workbooks must be set up manually which requires the user to put in a substantial amount of work. There are reporting tools such as charts, graphs, and tables but they can be complicated to use and again must be manually created by the user.

Strengths:

- Very affordable.

Weaknesses:

- Needs to be manually configured.
- All reports and ledgers must be built from scratch.

- Time intensive.

Strategic Partnerships

1) Shopify [14]

By partnering with Shopify, I could allow users to pull their Shopify sales and transactions directly into my application. This saves the user time as they would not need to input those transactions manually. The ledgers and reports would be automatically updated without any input needed from the user. Shopify would gain an advantage of brand awareness since it is incorporated into more third-party applications.

2) Amazon Business [15]

By partnering with Amazon Business, I could pull amazon transactions such as purchases or sales into my application automatically which would save the user time. The benefit for amazon is more brand awareness since their APIs would be incorporated into more applications.

3) Moneysoft Payroll [16]

Payroll is a feature that many accountants and small businesses require but is something that is not included in my application. If I could partner with a payroll specific software company by allowing users to integrate their payroll data into my application, then I could expand the features and flexibility of my application and reach a wider customer segment. I could also allow the partner payroll software to take data and transactions from my application to display on their software. For example, if the user wanted to see their business reports inside of the payroll application that they use my application to get the business's data.

4) Teller.io [17]

Teller.io is a company would allow me to connect my application to my user's bank accounts. This means that the user could pull their bank transactions into the application automatically which would save the user time and enhance my application. I could then build a bank reconciliation feature that would allow users to match the transactions in their bank account with the sales and expenses that they have entered.

5) Salesforce [18]

Salesforce is a customer relationship management software. If my application had access to that data, it could keep a record of customers and link each necessary transaction to a specific customer. This would give the user more information about their transactions when reviewing their ledgers. The alternative would be to build this functionality directly into my application but if customers already have that data on Salesforce, then it would be easier for them to just link it to my application instead of having to enter it again.

Revenue Model

Monetisation

After analysing the competition, I concluded that I should also follow the same monthly subscription model, but I will initially charge £5 a month since I don't offer as many features as my competition therefore my price should reflect that. FreshBooks is my closet competitor as they also cater to small businesses and their pricing is £12 a month [5]. Since they offer more features than me and are at a well-developed and defined stage, I will have to lower my price to compete. Once I have developed the application further by adding more standout features then I will increase the price close to £10 depending on my costs.

£5 is an affordable price that makes me competitive enough to be a good option for potential customers but also cover the costs of the application for now. It would make sense for me to reduce the price initially and as I add more features, I could increase the price accordingly. In the future I could also allow users to customize what features they want to use, and I could adjust the price accordingly. If I were to get an influx of customers right away, then my hosting costs will also increase meaning the price may need to change but I don't believe that is likely.

I could also offer referral discounts to provide an incentive for users to market my application for me and convince other people to use it. Both users will receive a discount for the first 3 months if they pay with a referral code. I considered giving free trials for first time users, however if I create tutorial videos on my website then users could have a better understanding of how my application works before they decide to commit to using it, meaning it may not be necessary.

Revenues & Expenses

Estimates Of Revenues

Since I plan to charge £5 a month for my application any user at a given time would generate £60 a year in revenue. During my first year, I would expect to have only a handful of users, but this could change depending on how effective my marketing campaign is as well as my product and pricing.

A strategy that I plan to implement is try and gain 5 customers every month in the first year as it allows me to gradually improve and refine my application to suit my customer's needs. My first-year business objectives are to breakeven and work on improving the product by getting customer feedback. If I grow gradually, it will help me to improve my application and better understand the market instead of being overwhelmed. Once my application is established, I will then switch my goals to maximise the number of customers I can get.

If I start with 10 customers, then by the end of the year I would have 75 which would generate £4,500 revenue for that year.

Estimates Of Expenses

The main cost for my application would be the cloud hosting on AWS or Azure since those are the two providers that I have experience using. Both providers offer a pay on demand service where I am only charged for the computing resources I use. This means that if I don't have many customers using my application then my fee will be reduced.

Figure 6: Azure vs AWS price comparison [6]

Instance Parameters	Azure Per-Hour Price	AWS Per-Hour Price
On-Demand / Linux / General Purpose / 4 CPUs / 16 GB Memory	$0.1670	$0.1856
On-Demand / Linux / Compute Optimized / 4 CPUs / 16 GB Memory	$0.1690	$0.1700
On-Demand / Linux / Memory Optimized / 4 CPUs / 16 GB Memory	$0.2660	$0.2660
On-Demand / Windows / General Purpose / 4 CPUs / 16 GB Memory	$0.5970	$0.8560
On-Demand / Windows / Compute Optimized / 4 CPUs / 16 GB Memory	$0.7260	$0.8340
On-Demand / Windows / Memory Optimized / 4 CPUs / 16 GB Memory	$0.8500	$0.9520

Figure 7: Azure Pricing Plan for Linux Virtual Machines [7]

Instance	Core(s)	RAM	Temporary storage	Pay as you go
B1ls	1	0.5 GiB	4 GiB	$3.7960/month
B1s	1	1 GiB	4 GiB	$7.5920/month
B1ms	1	2 GiB	4 GiB	$15.1110/month
B2s	2	4 GiB	8 GiB	$30.3680/month
B2ms	2	8 GiB	16 GiB	$60.7360/month

Figure 6 shows that Azure is a cheaper cloud hosting platform in general compared to AWS and as I plan to use Azure, I estimate costs of $7.592 a month according to figure 7. This is $91.104 a year but it is not fixed. As I start to grow and get more customers, I may need to upgrade my machine so this price could potentially increase depending on how many customers I gain during the year.

I will also be advertising on google search which is more expensive compared to Instagram, Facebook, and Twitter [21]. There are many different factors that affect google advertising costs such as industry and location but the average cost per click is between £0.88 - £1.77 [21]. Google allows me to set a daily limit on the cost which allows me to stick to a budget. I could invest £10 a month for advertising to start with which would cost £60 a year and analyse the return rate. If I am not converting customers, I could try a different advertising strategy.

Design & Implementation

Requirements

Account Features

- Users must be able to create an account.
- Users must be able to record the key business information. (Company name, address, telephone, base currency, financial year end date).
- Users must be able to log in.
- Users must be able to log out.

Recording Financial Data

- Users must be able to create journal entries to record their business transactions.
- The system must provide a list of default nominal accounts that the user can select when entering journal transactions.
- Users must be able to delete or edit journal transactions.
- Users must be able to record payments & receipts of the business.
- The system must generate unique references for each financial transaction.

Viewing Financial Data

- System must display all the ledgers used and their transactions and balances.
- System must be able to generate the Profit & Loss statement and the statement of financial position.

Overview

My application has been designed using a client-server model. My frontend makes API requests to backend to either create, read, update, or delete data. My backend is a REST API that will interact with my database which is built using SQL.

Frontend

Next.js

The frontend is built using Next.js which uses React.js and server-side rendering. React allows me to build the application using reusable components. Server-side rendering is when HTML is generated on the server and then sent to the client. This provides a faster loading time for the client, and it also improves search engine optimization for my website.

Tailwind CSS

I also used the Tailwind CSS framework to improve the speed at which I could build the UI and write CSS. Tailwind allows you to write CSS directly into you HTML by using utility classes. This saves time as you don't need to create separate CSS files for each HTML page but instead select the classes you need directly.

Redux.js

Redux is a state management tool that allows you to save data in a store so that any component from any page can access it. It makes it easier to pass data between components and pages in React.js without having to bubble up.

Backend

Node.js
Is a runtime that allows me to build my backend using JavaScript. This means I will only need to use one programming language to build both my front and backend. It also allows me to use many different packages from the node package manager such as Express.js and Sequelize.js.

Express.js framework
Will allow me to create a REST APIs server that will accept requests from the front end and return the data required.

Sequelize.js
Is the Object Relational Mapper that will allow me to create my database tables with relationships and query them.

Security

When researching the latest security techniques for web applications in industry, I decided to use JWT for authentication and authorization of users. Once a user has signed into my application, I will send them an encrypted token that contains their user information and credentials. Whenever a user communicates with my server, I decrypt this token to verify their identity and then modify their records as requested.

I decided to use Bcrypt for encryption of passwords on the database as it has not been broken and is very strong and slow encryption [22].

I intend to setup HTTPS in the future when my application is hosted on the cloud. HTTPS is the secure version of HTTP which is the protocol that is used to send data across the internet [23]. This means that whenever a user logs into my application or sends sensitive data it will be encrypted.

Database

The database that I use is PostgreSQL since it is the most suitable for the type of data I am inputting. A business's financial data is highly structured, so it fit well in a SQL database using tables. It would not be suitable for a key to value relationship used in no-SQL databases.

Schema

Transactions
The transactions table will be used to store each business transaction. Each transaction would be categorised as either a debit or credit using the entry type column and have a nominal account to add further detail to the type of transaction made.

Nominal Accounts
The nominal accounts table will be used to categorise each transaction. Some example nominal account names will be stationary, fuel, sale of assets, income. They will have a reference number to further categorise each transaction for reporting purposes. For example, all nominal accounts referenced 5000 to 6000 are expenses. This also allows the possibility for the user to create their own nominal accounts in the future as they could create a custom name and use a reference number within the specific range to indicate what type of category the transaction would be.

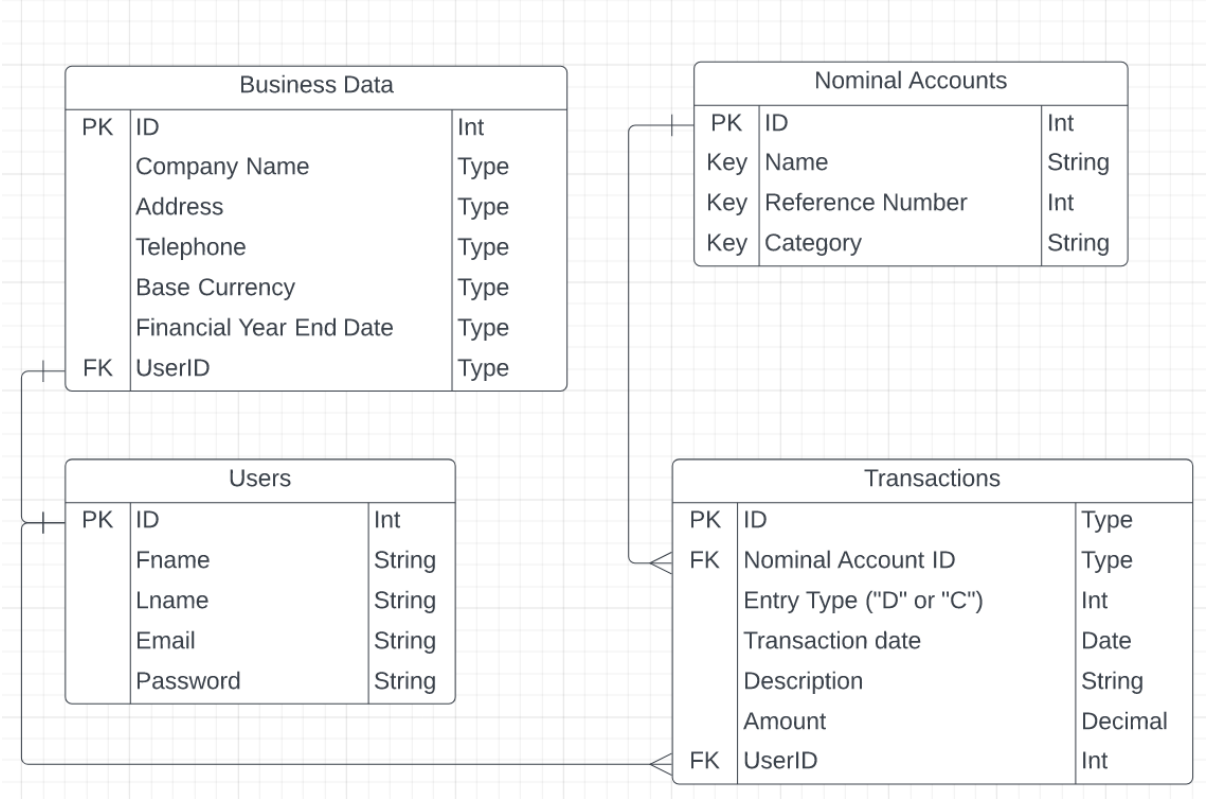
Business Data

This table will store all the general data about the company. Such as

Users

This table is to store the information about the user such as email and password. This table will be queried during authentication.

Entity Relationship Diagram



Results & Discussion

If I analyse my project against my aims and objectives, then I believe I have achieved the goals that I set out for myself as I have implemented each feature in the requirements and completed my objects listed above.

However, If I analyse the project from a business perspective then I believe that more can be done on the software itself and that the application should be expanded. I believe that the customer may appreciate more automation of the bookkeeping process as many of my competitors have been implementing such features for their users and therefore, I can compete with them if I also provide similar features. I believe that partnering with other businesses previously listed would add extra functionality to my application and make it more well-rounded to suit the needs of more diverse customer segment. Partnering with a payroll software company or even implementing it myself in the future can allow me to target medium sized businesses as customers.

User Interface

Figure 8: Dashboard page

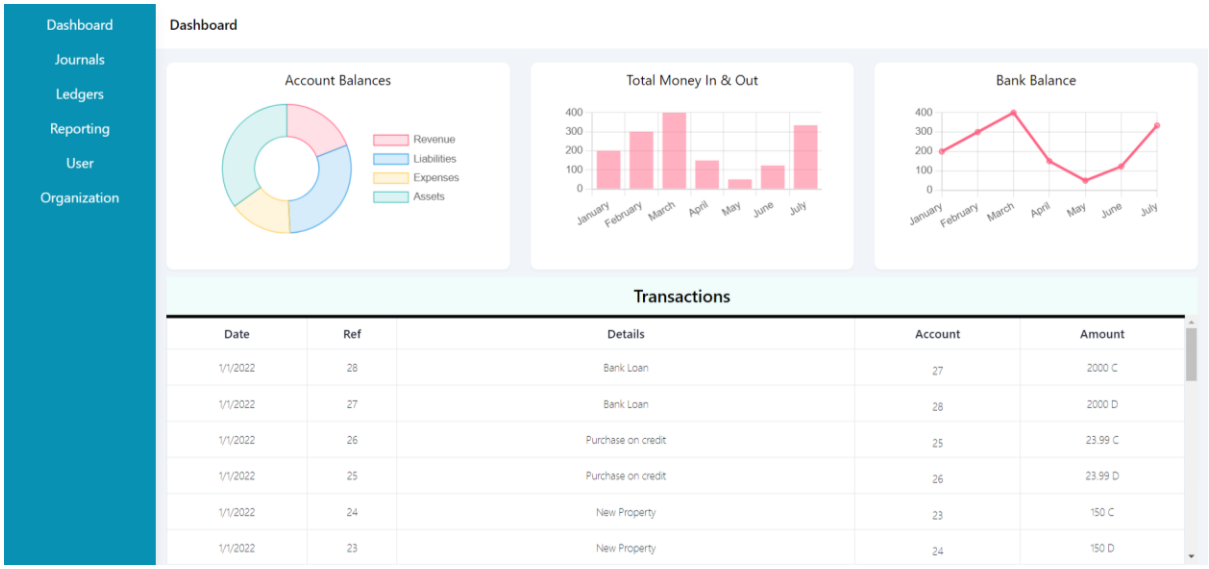


Figure 8 shows the dashboard page which provides three reports to give a general overview of the financial standing of the business. You can also view the most recent transaction in a table below. If a user clicks on the account under transactions, they are navigated to the ledgers page where the correct transaction is shown.

Figure 9: Account balances report

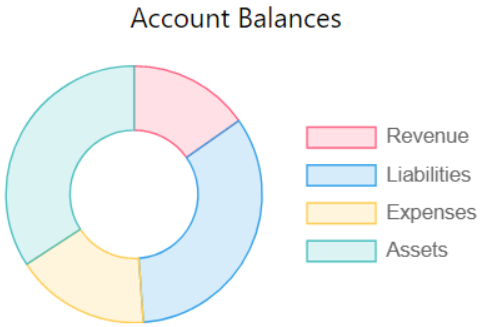


Figure 10: Account balances report filter

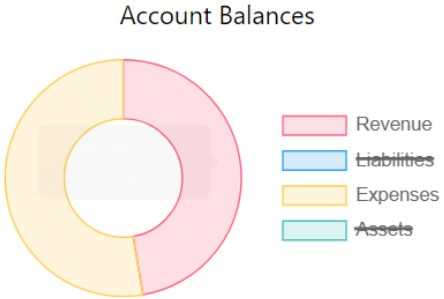


Figure 9 and 10 shows the sizes of the different subgroups that make up the profit and loss and balance sheet for the current year end. This allows the user to compare each of them at a glance and to get a general overview of financial standing of their business. It works by looping through each ledger and adding the balance to the correct subgroup.

Figure 11: Cash flow report

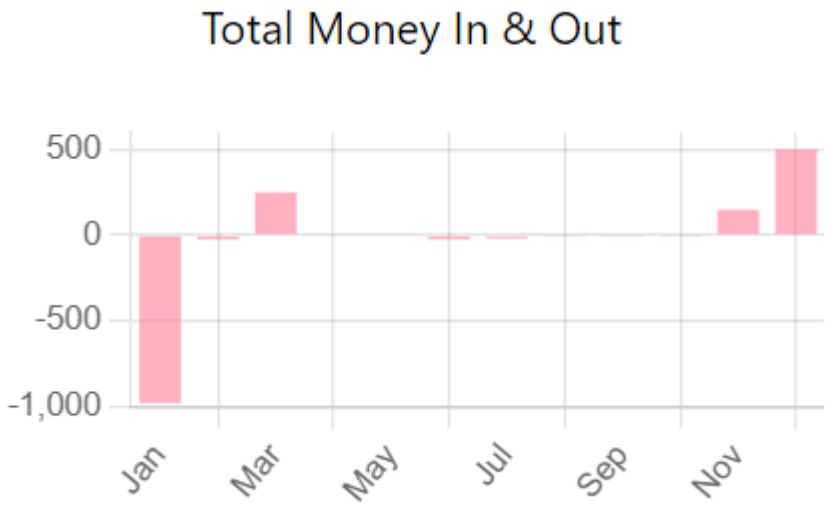
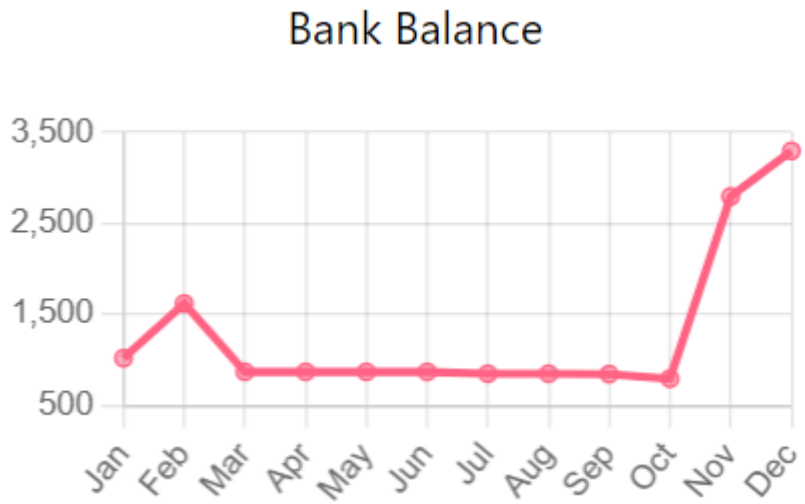


Figure 11 shows the final balance of the cash flow report for each month. This allows the business owner to know if they are going to run out of money. I display it in a bar chart format instead of a static report like that seen in the profit and loss statement to make the data more meaningful as the user can see their progress over time.

I had to research into cash flow and what subcategories of ledgers would be responsible for displaying this data [26]. I tally up the balances for the income ledgers and subtract it from the all the expenses ledgers for each month.

It works by filtering ledgers to use only the ledgers responsible for cash flow. I loop through the correct ledgers 12 times which reflects each month in a year and filter again based upon the month. The balances are calculated, and the values are updated in the bar chart by either adding or subtracting, depending on the ledger and balance.

Figure 11: Bank balance report



This shows the current bank balance for each month and allows the business owner to examine it over the year. The user can see how much cash is available to be used at any given time. It works by looping over the bank ledger and appending the balance at each month.

Figure 12: Ledgers page

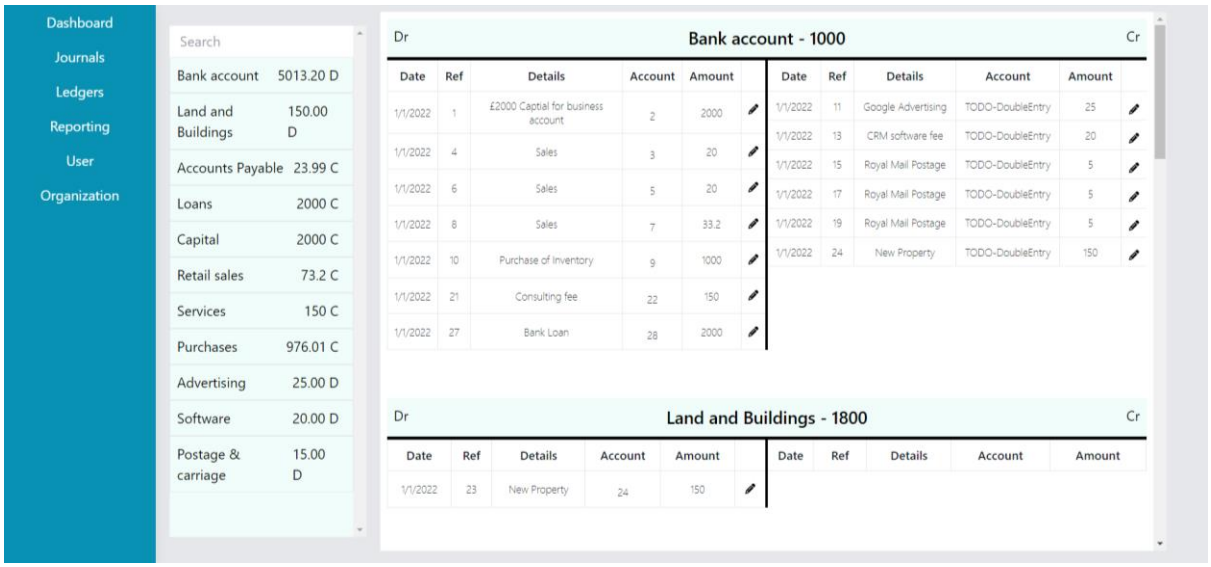
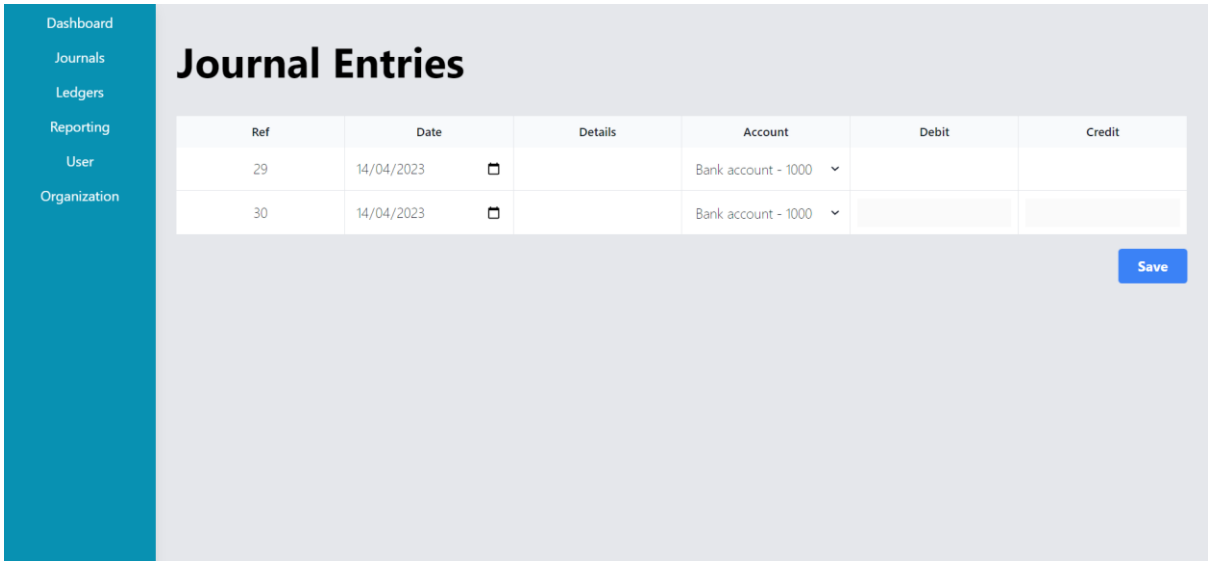


Figure 13: Ledger filter

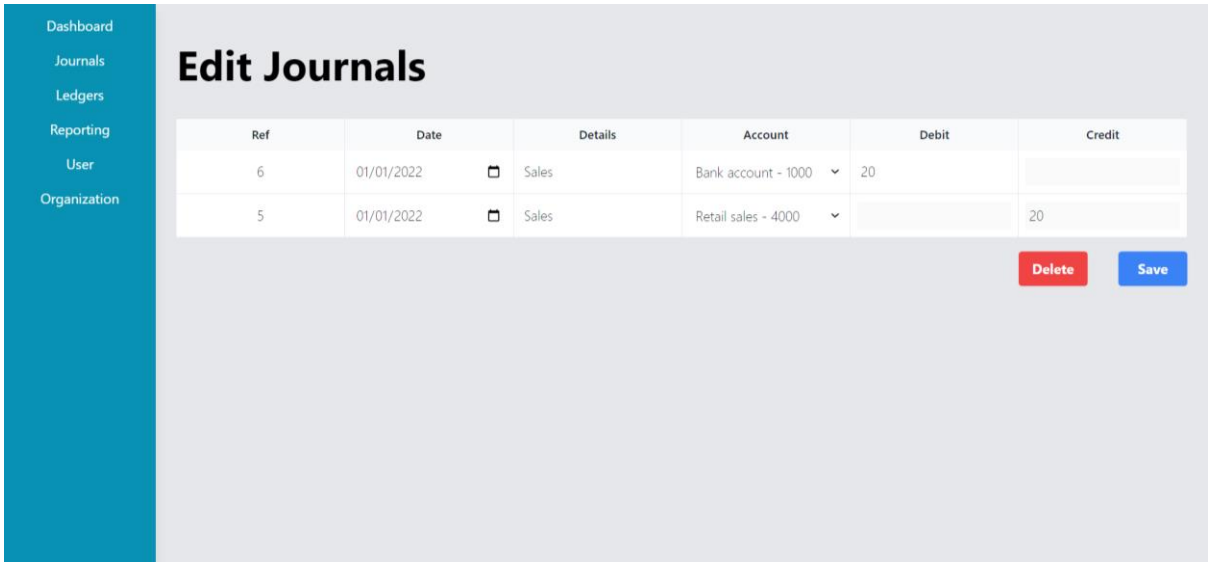
The ledgers page is where the user can see all the different business transactions organized into simple T-account ledgers. This is a familiar sight for those who understand the fundamentals of bookkeeping. The user can search for a specific account using the search filter and select it to view the corresponding ledger as seen in figure 13. This saves the user time from scrolling. A user can edit an entry by selecting the pencil icon which will navigate him to the Journals page where the double entry information will be set by default. The transactions inside the ledger are sorted by date and by clicking on the account number you will be navigated to the matching double entry.

Figure 14: Add journals page



The Journals page is used to record new business transactions. The reference number is automatically incremented and the amount can only be entered into either the debit or credit column to maintain the double entry standard. The second transaction's details and amount inputs are disabled but its value will be matched to transaction one to make sure the rules of double entry are in place and that all ledgers will balance. When a user selects the save button the inputs are cleared indicating that the submission was successful.

Figure 15: Edit journals page

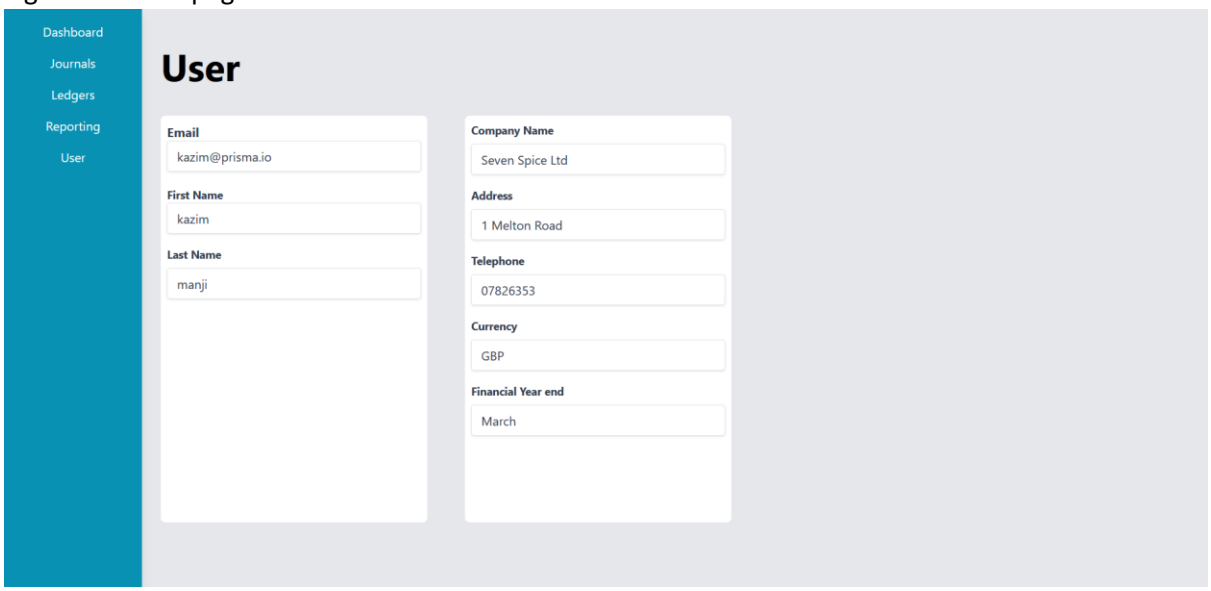


The Journals page can also edit existing entries or delete an entry. The user can only do this if they select the pencil icon in the ledgers page then they are navigated here with the inputs prepopulated. The user can choose to delete a record or save their changes. After the transaction is edited, the user is navigated back to the ledgers page to the correct ledger.

Figure 16: Reports page

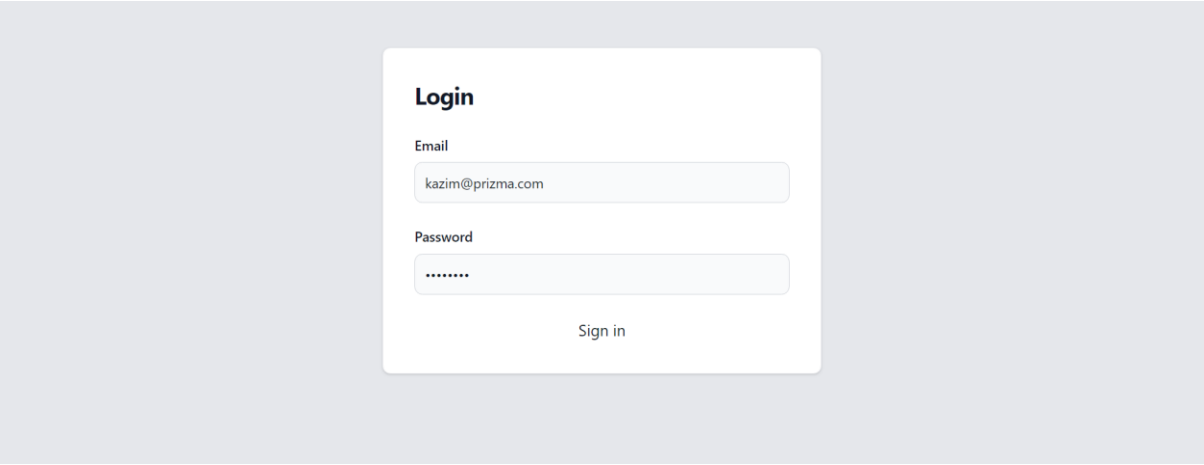
The reporting page will generate and display the Profit & Loss and Balance Sheet reports to the user. These reports are part of the accounting process. The Profit & Loss statement summarizes the revenues, costs, and expenses while the Balance Sheet reports the company's assets, liabilities, and equity. The ledgers are organized by financial statement and then by subgroup and then the balances are calculated.

Figure 17: User page



The User page shows the company information and the current user's information. The user will be able to edit this information in the future and change their current financial year end.

Figure 18: Login page



This is a simple login page that allows the user to access the application. The password is not visible for additional security. Once a user logs in, they are redirected to the dashboard page.

Figure 19: PostgreSQL database schema

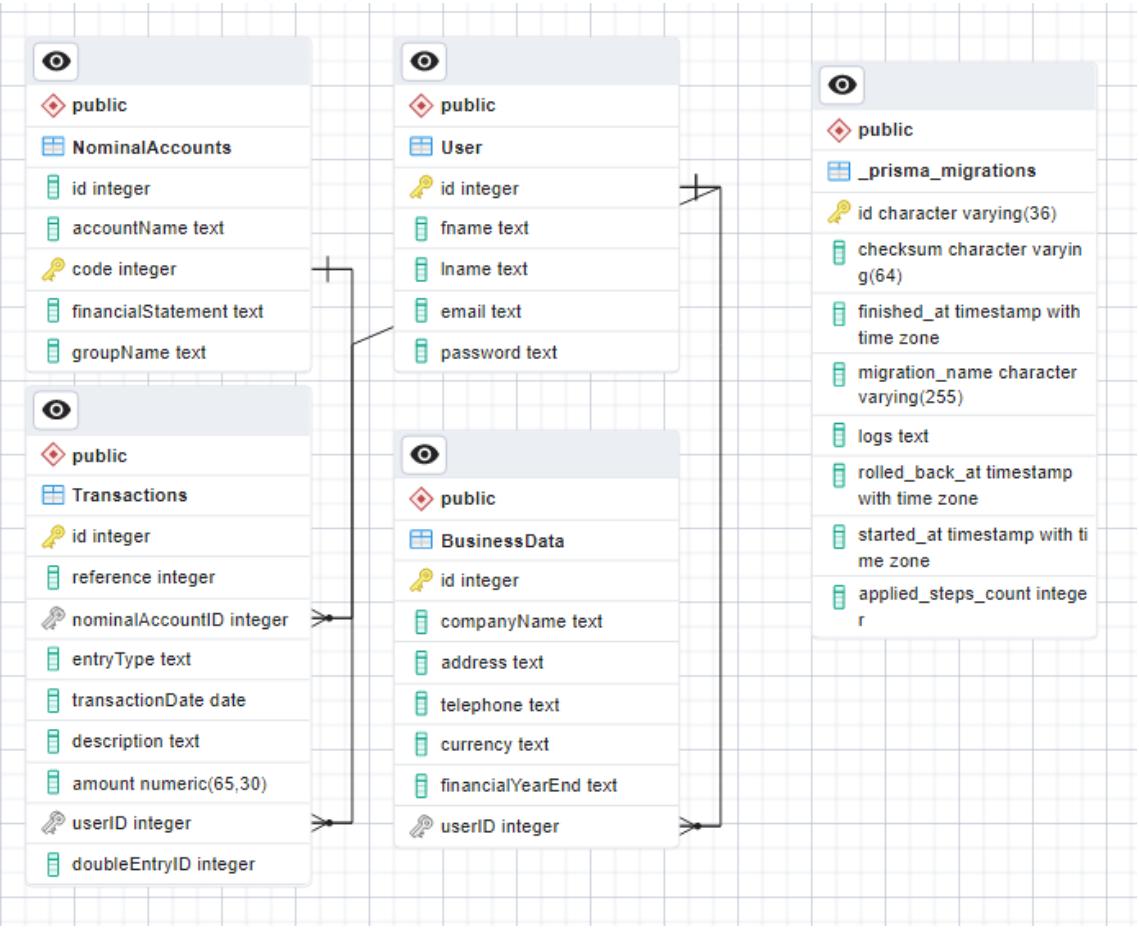


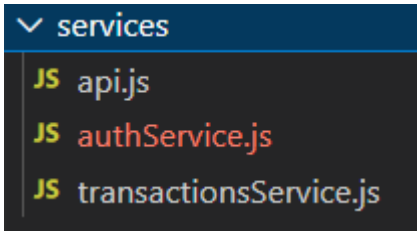
Figure 19 shows the final database that was built in PostgreSQL from my designs.

Project Structure

Frontend

Services

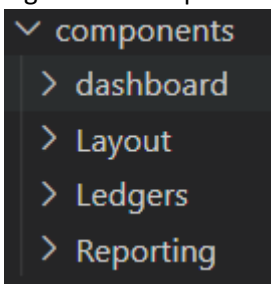
Figure 20: Services folder for API requests



I used the Axios library to make API request from my frontend to my backend. These requests are made up of authentication requests such as login, register, logout which are stored inside authService.js and transaction requests such as create, edit, delete transactions which are stored inside transactionService.js. I initialized the Axios instance inside api.js and set the JWT authorization header by reading from local storage if the user has logged on. I then created functions for each request in separate files for code readability.

Components

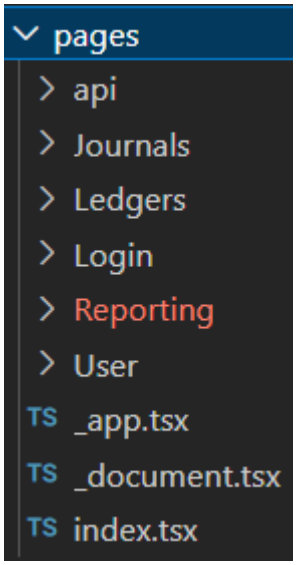
Figure 21: Components folder



I have created React.js components that are used by different pages and stored them in this folder. My pages can import these components when needed.

Pages

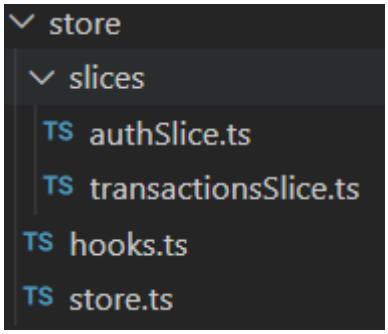
Figure 22: Pages folder



Under each subfolder is the React component that makes up each page in my website. Each folder here is associated with a different route for example the component returned in the /Journals route will be the one inside the Journals folder here. Some of these pages import components from the component folder. Index.tsx is my dashboard landing page.

Redux Store

Figure 23: Redux folder

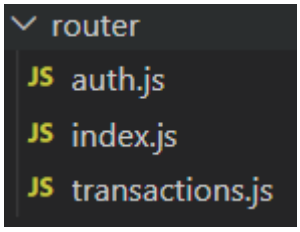


The redux store is used to persist the application state between pages. It is like a database and allows me to access and mutate data from different pages and components. The redux instance is initialized inside store.ts and the application specific services are inside the slices folder. When I make requests using Axios the data is stored here. Whenever I refresh a page, my redux store will check if the data already exists in local storage and save it. This prevents my application from loosing all data upon refresh.

Backend

Router

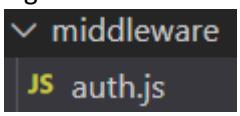
Figure 24: Router folder



My server listens for requests coming to it using Express.js. These listeners are implemented in the router folder in different files for readability. They call middleware functions as well as controller functions.

Middleware

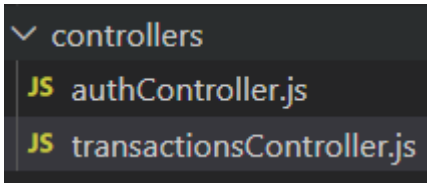
Figure 25: Middleware folder



Middleware functions can be chained inside of my Express.js listeners. Once a request is received, I can pass it on to multiple middleware functions before sending it to my controller. The auth.js middleware function is used inside my routes so that my server will check the authenticity of the request before processing. The function will decrypt the JWT token found in the request header and get the user's information and pass this information along with the original request to my controller. If the token is expired, the request will be rejected.

Controllers

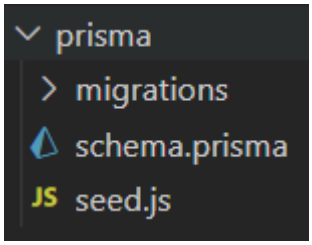
Figure 26: Controller folder



The controllers will perform CRUD operations on my database as requested by the client. They are encapsulated in try and catch statements to prevent my server crashing if there is ever an issue with a request and instead return an error message. This was challenging at times as I had to learn about concurrency in JavaScript and how to use "async", "await" and "then" statements allow database operations to be performed asynchronously. This meant my server could wait for database operations to finish before moving on to the next line code.

Prisma ORM

Figure 27: Prisma folder



Prisma is the object relational mapper, and this folder contains all the code that allows me to create database tables and interact with them. My controller will import code from here to interact with the database. The seed.js file allows me to populate my tables with test data. The schema defines my database tables and their relationships. The migrations folder keeps track of any database migrations and changes occur.

Testing

Figure 28: Postman API test for login route

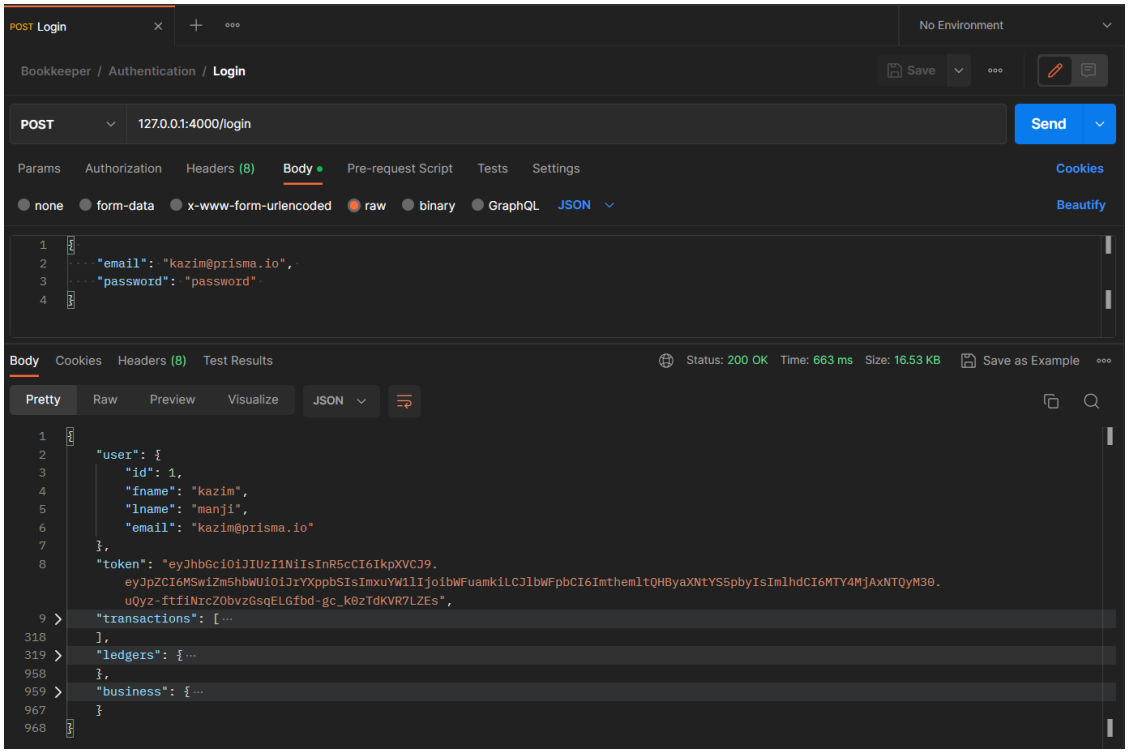
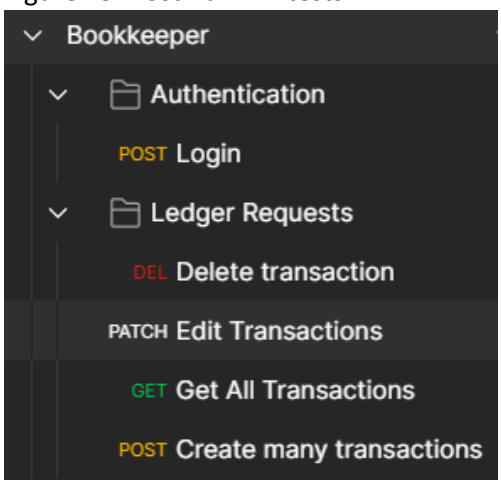


Figure 29: Postman API tests



I performed API testing using a software called Postman where I was able to send requests to my backend and view the return JSON objects [24]. Figure 28 gives an example of the test for my login route. I wrote tests for all the communications between my frontend and backend as shown in figure 29. While writing my code I also utilized try and catch statements that allowed me to return any errors during my development. This meant that my console would be displaying any errors if they occurred. Using try catch statements also meant that my server would not crash if an error occurred which meant a user could continue to use the application.

Critical Appraisal

Summary of work done & Critical analysis of work done:

I believe that the original aims of the project have been achieved as I have bult a full stack web application that allows users to record their business transaction following the double entry bookkeeping rules. The users have the freedom to select which ledgers they want to make records in and can delete or edit those records, meaning they have full control over their accounts. The users can use the overview reports in the dashboard page but also view their Profit & Loss statement and Balance sheet.

I believe the UI is simple to use which is part of my unique value proposition, but it could be improved by writing a user guide on how to use my application. Tutorial videos could also be created and would also act as advertising content if posted to social media. I believe the ledgers page could have more filters to manage each transaction. Further testing could be carried out such as compatibility with different browsers. Usability testing would also have been beneficial for allowing me to get direct feedback from users.

There are some enhancing features that could be included to improve the application and make it more user friendly. For example, If the user could download the reports as a PDF file, it would allow the user to share those documents with the necessary parties in the organization. I could also improve the dashboard reports by building a regression model to be able to predict the users cashflow for the future [27]. By enhancing my application through new features that were not listed in my requirements I would add more value for the customer and would be more likely to get sales.

Discussion on Social, sustainable, commercial, economic context

The benefits for businesses would be that they would be able to use my application to record their transactions and better understand their financials. They can examine the reports to get an overview on their expenses better manage their finances by understanding their fixed and variable costs. By examining their transactions, they can make estimates on the future and make better management decisions.

Hacking and data loss is potential risk as this sensitive business data could be stolen by hackers. The use of any new software also means that users must be trained and get used to doing things different. My business is intended solely for commercial purposes, but it can be used for learning by those new to accounting. They may better understand bookkeeping and practice making journal entries to build a profit and loss statement or balance sheet.

Since I will be competing with other businesses in my industry, I will be pushing them to become more competitive and could drive innovation to even create new products or features [25]. They may need to hire new developers to achieve this which creates jobs. New markets could be created through innovation which could grow the economy by creating new job opportunities and revenue streams [25]. Generally, competition is good for consumers as it causes businesses to lower prices and build better products [25].

Personal development

Throughout the project I have been able to improve my understanding with JavaScript and learnt how to incorporate TypeScript for the first time which provided my frontend with type safety. I have used new frontend frameworks such as Next.js and Tailwind.js to build my application in a component-based manner. I had researched into web security to learn about HTTPS, encryption algorithms and authentication using JWT. I improved my problem-solving skills as I ran into many issues when trying to work with different libraries that I had not used before. I would carry out research and come up with workarounds to resolve those problems allowing me to apply myself fully. I have gained a better understanding over asynchronous code and the difference between lazy and eager loading of database entities to be used in my backend. Overall, I have a better understanding of full stack web applications and the different methods to build them.

I learnt how to analyse competitors and build a lean canvas to better understand the components that make their business. I also applied this to my own business and was able to understand how to differentiate myself my developing a unique value proposition. I also learnt how to make estimates of expenses and revenues and plan different marketing strategies to suite my business. I learnt how to identify potential business partners and understand their benefits. I learnt how to specify my customer segment and research where to find potential customers and potential advertising platforms to target them.

Conclusion

Overall, I believe that I have met the project aims and objectives and completed the requirements laid out at the start of the project. I have built an application that allows bookkeepers for small businesses to input and track their business transactions for accounting purposes. For example, their income, expenses, assets, liabilities. The application allows them to get an idea of where their

business stands financially by generating financial statements and provides a dashboard with reports so they can see their progress over time.

It is competitive through its simplicity as it allows users to input financial data through journal entries and displays the transactions in T-account ledgers. Users have full control over the accounts as they get to choose which ledgers to make entries into.

I have carried out market analysis, developed marketing strategies, and created a business plan that I can implement to make my product more likely to succeed in the future. The business development undertaken makes my project dynamic and versatile as I have taken effort to understand another dimension of software development beyond just the design and implementation.

I analysed my competitors and understood their unique value proposition and the features that they offered to their customers. I built a lean canvas to better understand all the factors that effected my business and industry. I managed to implement my design for both the frontend and backend using the technologies that I had intended as I have built this application using Next.js, JavaScript, Node.js and PostgreSQL as the core technologies. I was able to research and design a database that follows accounting rules and the double entry system and carried out testing using Postman as well as using sample data seeded into my SQL database using the Prisma.

References

[1] “Business population estimates for the UK and Regions 2022: Statistical release (HTML),” *GOV.UK*. [Online]. Available: https://www.gov.uk/government/statistics/business-population-estimates-2022/business-population-estimates-for-the-uk-and-regions-2022-statistical-release-html#:~:text=common%20legal%20form.-,At%20the%20start%20of%202022%3A,proprietorships%20and%2095%2C000%20ordinary%20partnerships. [Accessed: 08-Apr-2023].

[2] “Industry market research, reports, and Statistics,” *IBISWorld*. [Online]. Available: https://www.ibisworld.com/united-kingdom/market-size/accounting-auditing/. [Accessed: 08-Apr-2023].

[3] auth0.com, “JSON web tokens introduction,” *JSON Web Token Introduction*. [Online]. Available: https://jwt.io/introduction. [Accessed: 10-Apr-2023].

[4] “Accounting software – do beautiful business,” *Xero*. [Online]. Available: https://www.xero.com/uk/. [Accessed: 10-Apr-2023].

[5] “Invoice and accounting software for small businesses,” *FreshBooks*, 04-Jan-2023. [Online]. Available: https://www.freshbooks.com/en-gb/. [Accessed: 10-Apr-2023].

[6] P. E. Yifat Perry, “Azure vs AWS pricing: Comparing apples to Apples,” *NetApp BlueXP*, 10-Sep-2020. [Online]. Available: https://bluexp.netapp.com/blog/azure-vs-aws-pricing-comparing-apples-to-apples-azure-aws-cvo-blg#H_H8. [Accessed: 11-Apr-2023].

[7] “Pricing - linux virtual machines: Microsoft Azure,” *Pricing - Linux Virtual Machines | Microsoft Azure*. [Online]. Available: https://azure.microsoft.com/en-gb/pricing/details/virtual-machines/linux/#pricing. [Accessed: 11-Apr-2023].

[8] R. Peace, “Google ads vs. Facebook ads: Which is best?,” *AgencyAnalytics*, 02-Sep-2022. [Online]. Available: https://agencyanalytics.com/blog/google-ads-vs-facebook-ads. [Accessed: 12-Apr-2023].

[9] G. Author, “8 steps to a [truly] traffic-generating seo strategy in 2022,” *WordStream*, 26-Jul-2022. [Online]. Available: https://www.wordstream.com/blog/ws/2021/03/05/seo-strategy. [Accessed: 13-Apr-2023].

[10] Jeel Patel, “Django vs express: 2 best web frameworks compared in 2023,” *Monocubed*, 03-Mar-2023. [Online]. Available: https://www.monocubed.com/blog/django-vs-express/#:~:text=Django%20is%20an%20advanced%20Python,js. [Accessed: 10-Apr-2023].

[11] “Protecting against SQL Injection,” *Hacksplaining*. [Online]. Available: https://www.hacksplaining.com/prevention/sql-injection#:~:text=Object%20Relational%20Mapping&text=ORM%20tools%20often%20mean%20developers,parameterized%20statements%20under%20the%20hood.&text=Code%20like%20this%20is%20safe%20from%20SQL%20Injection%20attacks. [Accessed: 10-Apr-2023].

[12] “What is content marketing?,” *Content Marketing Institute*. [Online]. Available: https://contentmarketinginstitute.com/what-is-content-marketing/. [Accessed: 10-Apr-2023].

[13] “Smart, simple accounting software: QuickBooks UK,” *Smart, Simple Accounting Software | QuickBooks UK*. [Online]. Available: https://quickbooks.intuit.com/uk/. [Accessed: 10-Apr-2023].

[14] “Start and grow your e-commerce business - 3-day free trial,” *Shopify*. [Online]. Available: https://www.shopify.com/uk. [Accessed: 10-Apr-2023].

[15] “Procurement & Wholesale Solutions,” *Amazon Business*. [Online]. Available: https://business.amazon.co.uk/. [Accessed: 10-Apr-2023].

[16] “Moneysoft payroll software and bookkeeping software,” *Moneysoft*, 17-Mar-2022. [Online]. Available: https://moneysoft.co.uk/. [Accessed: 10-Apr-2023].

[17] “The API for your bank account,” *Teller*. [Online]. Available: https://teller.io/. [Accessed: 10-Apr-2023].

[18] “The customer company,” *Salesforce*. [Online]. Available: https://www.salesforce.com/uk/. [Accessed: 10-Apr-2023].

[19] “Xero Certifications,” *Xero Central*. [Online]. Available: https://central.xero.com/s/certifications. [Accessed: 10-Apr-2023].

[20] “Freshbooks vs QuickBooks,” *HubSpot*. [Online]. Available: https://www.hubspot.com/comparisons/freshbooks-vs-quickbooks. [Accessed: 10-Apr-2023].

[21] “The Complete Guide to cost of advertising on google,” *Award Winning eCommerce Agency London, UK*, 14-Jun-2022. [Online]. Available: https://www.rvsmedia.co.uk/articles/the-complete-guide-to-cost-of-advertising-on-google/. [Accessed: 10-Apr-2023].

[22] “What is the best hashing algorithm?,” *Code Signing Store*, 07-Apr-2022. [Online]. Available: https://codesigningstore.com/what-is-the-best-hashing-algorithm#:~:text=To%20protect%20passwords%2C%20experts%20suggest,your%20best%20hashing%20algorithm%20choices. [Accessed: 10-Apr-2023].

[23] S. S. L. S. Team, “What is HTTPS?,” *SSL.com*, 13-Oct-2021. [Online]. Available: https://www.ssl.com/faqs/what-is-https/. [Accessed: 10-Apr-2023].

[24] *Postman*. [Online]. Available: https://www.postman.com/. [Accessed: 10-Apr-2023].

[25] “The benefits of competition,” *Autoridade da Concorrência*. [Online]. Available: https://www.concorrencia.pt/en/benefits-competition#:~:text=By%20competing%20with%20each%20other,and%20well%2Dbeing%20for%20society. [Accessed: 21-Apr-2023].

[26] PricewaterhouseCoopers, “Preparing a cash flow forecast: Simple steps for Vital Insight,” *PwC*. [Online]. Available: https://www.pwc.com/gx/en/services/entrepreneurial-private-business/small-business-solutions/blogs/preparing-a-cash-flow-forecast-simple-steps-for-vital-insight.html. [Accessed: 27-Apr-2023].

[27] [1] W. G. Klerck, (PDF) use of regression analysis to forecast cash flows - researchgate, https://www.researchgate.net/publication/335309661_Use_of_regression_analysis_to_forecast_cash_flows (accessed May 3, 2023).