Online Banking Application Using A MERN Stack

Ultan Kearns

B.Sc.(Hons) in Software Development

January 15, 2020

Final Year Project - Banking Application using MERN stack

Advised by: Dr. Dominic Carr
Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

About This Project

Analysis of This Project: This project was designed by me for the module Applied Project & Dissertation with the purpose being to complete an online banking system using a MERN(Mongo, Express, React & Node) stack. The project should allow user to login, view statements, takeout loans, perform transactions and will show the user there monthly and yearly expenditures using graphs. The project will also be secure and utilize user accounts using a mongo Database to ensure that the users account is secure.

The main purpose of this project is to utilize the MERN stack to provide a full and rich user experience and to provide a secure, intuitive and polished online banking system. The project will also utilize Python scripts to perform statistical analysis on user expenditure and income and will provide an estimate of how much money the user should have for the month based upon previous monthly expenditure.

This project was designed to be a stand alone application where a user can perform all their banking needs without any other software. The user should be find the UI intuitive and the features helpful.

This project will link many disparate technologies together for the purpose of providing the user with the features they need. I plan to use this project to show the skills I have attained during my course and to learn a new framework(React). I also plan to improve and cultivate my skills using new technologies such as various Python libraries and React. This project is stored on a GitHub repository the link to which can be found in the appendix.

Authors: My name is **Ultan Kearns**, I am a fourth year student at GMIT. I have never used React or LaTeX before but I plan to learn a lot about these technologies during the course of this project.

Chapter 1

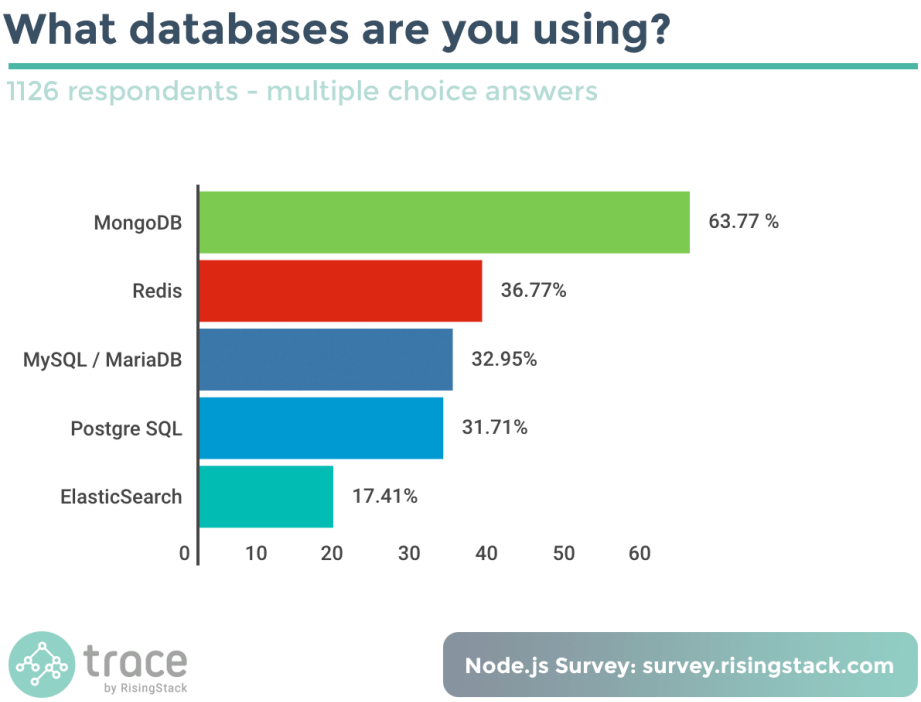
Introduction

1.1 Why A MERN stack?

There are many reasons why I have chosen to use a MERN stack for this application the main one is because it provides a framework for a full stack application. MERN stands for **Mongo - For the database backend(Hosted on Mlab, all user info is stored here), Express - A web application framework host a server in a relatively short time compared to other methods , ReactJS for asynchronous JS & Node - For the server and also includes a package manager to install a variety of packages to provide certain functionalities.** [1]

1.1.1 Mongo

As you have read above the MERN stack is very powerful in creating a full-stack application when used correctly. The reason I chose the database Mongo is because it offers an open source alternative to MySQL and other proprietary databases [2] and many companies seem to be migrating to it because it is open source and in some cases may offer better performance than other databases.



[3] Mongo is easy to learn as it stores it's data in JSON format and is schema-less, that is the user defines their own schemas. I have also used it for 2 other projects in Angular and find it an easy database to use in relation to projects.

[4]

1.1.2 Express

I chose to use Express because it is a very useful framework when creating a server. It is very helpful when pulling data from the server and displaying it to the client it is also very scalable and offers good security when used properly. It is also very easy to setup and can be connected to the MLab database in minutes upon starting the project. I have also used this server before and have a bit of experience with it and found it very helpful in the projects I used it for which were a messaging forum and an E-commerce application both using MEAN(Mongo,Express,Angular & Node) stacks.

[5]

1.1.3 React

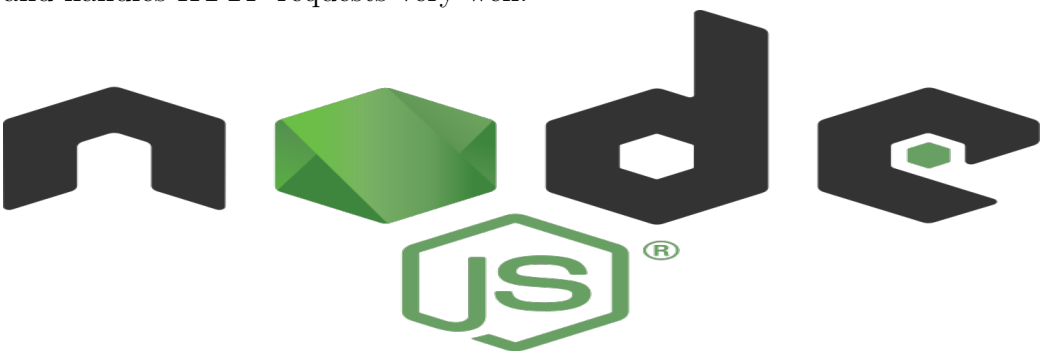
I also chose this project because I have zero experience in ReactJS and since it's such an up and coming framework I decided on it for this project. It also offers many features and libraries which are desirable when programming a user interface to ensure the user finds the application intuitive and easy to use. ReactJS also utilizes modular programming for components which makes designing web applications far easier than just using HTML, CSS &

Javascript. React is also very fast at rendering pages so that the user will not experience a major delay in accessing information.

[6]

1.1.4 Node JS

The reason I chose Node JS was because it offers a great package manager which can be used to improve productivity on my part and also the applications security, UI, UX and various other aspects of the application. I debated using Yarn for this project but settled on Node because it was already installed on my laptop. Node is a very useful tool when developing full-stack applications and was very helpful in installing tools and libraries for react. Node is also very good for getting a server up and running in a few minutes and handles HTTP requests very well.



[7]

1.2 Why a banking application?

1.2.1 Explanation of Why I Chose This Project

A banking application is broad in scope and offers many questions to the developer such as how can I optimize load times and how can I ensure user data is secure. I think questions such as these offer the potential for growth in the areas of design and problem solving - which are two of the most major skills a software developer can possess. I feel that an application such as a multi-user banking system can be beneficial to my career and help to improve my skills as a developer. Banking applications also have the potential to offer a wide array of features and are ubiquitous in the real world, think major banks such as AIB & Bank of Ireland. Banking systems also offer a broad range of problems such as security, design and usability.

1.2.2 My Contention With Existing Online Banking Systems & How I Plan To Improve Upon Them

The current generation of online banking systems or at least the ones I have used tend to have a variety of problems. These problems are glaringly obvious to most people and the main problems include but are not limited to: usability, appearance, lack of personalization & lack of information given to user. I will how I plan to solve each of these problems below:

- **Usability:** I aim to provide an intuitive and cutting-edge user interface using the latest react libraries to provide the user with an easy to understand banking application. All features will be easily navigated to using a navigation bar and I will aim to make features as obvious as possible to the user.
- **Appearance:** The UI of the modern day online banking system tends to be absolutely depressing. I aim to reduce this by adding in a responsive UI and to offer the user a vibrant online banking experience.
- **Lack of personalization:** I aim to make this application very personal to the user by adding in personalized expenditure charts and giving the user a unique and inimitable banking experience.
- **Lack of information:** Modern banking applications sometimes display a lack of information given to the user. I plan to solve this by

offering the user expenditure charts, reports and also by sending automated emails to them when their account balance falls below a user specified number.

1.3 Requirements

Below I will include the requirements for this application and expand upon them.

- **Multiple Users:** This application must allow multiple users to execute simultaneous banking and user sessions must be independent of each other.
- **Secure:** Database information must be encrypted
- **Accurate:** All statments and user information must be accurate.
- **Login/Logout** The user must be able to login and logout
- **information**: The user must be able to view all information related to them (eg: statments, withdrawl dates etc.)
- **Graphs/Charts**: The banking app must display expenditures and credit in graphs and charts generated from python scriptss
- **Emails** The application must be able to email the user a forgot password if they forgot their password and alos be able to email the user if budget controls are turned on.
- **Register** The user must be able to register new accounts using an email and password also ensure that it cannot be an email that already exists.
- **Delete account** The user must be able to terminate their account and all information that exists about the user must be purged from the server.
- **Change information** The user must be able to update all their personal information in relation to their account except obviously banking statments and balances.
- **User sessions** The application must use cookies to maintain a user session and ensure that the cookies do not last more than a specified timeframe max of a day.

1.4 Outline of chapters

Below I will outline the chapters that my Dissertation is broken up into and give a brief outline of each one.

1.4.1 Context

In Chapter 2 I will discuss the context of my project and how online banking applications have affected the modern age and traditional banking. I will research how online banking came about and how it is useful for the consumer as well as the bank.

1.4.2 Methodology

In Chapter 3 I will discuss the methodology I followed and how it affected my project and productivity. I will also discuss my how I planned to complete the project and discuss the methodologies I utilized in completing this project. I will discuss why I chose these methodologies and give the reader an insight into how this application was developed.

1.4.3 Technical Review

In chapter 4 I will discuss the technical aspects of this project and discuss how they impacted the development of this project and why they were implemented. I will discuss the MERN stack in more detail and how it was utilized to create a full-stack online banking application.

1.4.4 System Design

In Chapter 5 I will explain the architecture and design of this project. I will use graphs and diagrams to explain how the application is designed and how it will function when deployed. I will also present some of the code I used and how it is used to perform various functions of the banking application.

1.4.5 System Evaluation

In chapter 6 I will analyze the finished product. I will test the system and evaluate if it is up to standard and meets all the requirements I have specified. I will test if the scalability, security and the UI to ensure that the user is provided with the features outlined in the requirements section.

1.4.6 Conclusion

In chapter 7 will briefly outline what I learned from this project and highlight all my findings from previous sections. I will also discuss the impact the project had on my skills as a software developer and how it helped me to grow as a developer. I will also discuss what I would do differently if I had to do the project over again.

1.5 Structure of Project

The github project contains two branches feature and master I use the feature branch to test all new code and ensure that it works properly before merging it with the master branch. The git repository contains two folders banking-app and dissertation, the banking app folder contains the main project and the dissertation contains all materials relating to the dissertation. In addition to the two folders I also have a sprints.md file which contains information about all my sprints as I decided to use the agile methodology during this project, I also have a usefulresources.md file which contains a list of useful resources which helped me throughout the course of this project. The final file is the README.md this will contain a brief intro to the project as well as information on running the application.

Chapter 2

Context

2.1 Project Objectives

- To provide safe & secure online banking
- To provide an intuitive UI that can be easily navigated by the user
- To provide user generated statistical analysis of expenditure
- To provide a multi-user server and banking service
- To provide a RESTful API to the banking service(client/server totally independent ,stateless environment, caching)
- To provide a scalable application
- To provide user with security using encryption for the mongo database
- To limit loadtimes of traditional online banking eg. 365 Online Banking and others

2.2 Online Banking

Online banking is very relevant in today's modern world, the ease of access which the digital age allows us has led to a significant amount of banking being done online. Think of how many inconveniences have been shed away by the advent of online banking, no longer do people have to wait in line for ages at the bank to take out loans or to view their statements as all this can be done online or with the help of computers. Every modern bank now has some form of website which allows their customers to do all their daily

tasks related to banking online. In this chapter I will discuss the advantages, disadvantages and overall affect of online banking.

2.3 History of Online Banking

2.3.1 Definition of Online Banking

The definition of online banking includes any form of electronic payment systems that allows the customer or business entity to conduct financial transactions via a financial institutions website or application.

2.3.2 The Beginning

The metamorphosis of the old brick and mortar banks to the click and mortar banks of today all started as early as the 1980s [8]. The earliest version of the online prescence of banking took place in none other than New York City, New York, USA. Citibank, Chase Manhattan, Chemical Bank and manufacturers Hanover became the first banks to introduce on online banking system in 1981.[8] Then in 1983 the Bank of Scotland became the first bank in the UK to offer an internet banking service which was called Homelink to UK customers.People had to use their phones and their televisions to manage their online banking as computers where not as ubiquitous at this time [8] It was not until 1994 that a bank in the USA called the Stanford Federal Credit Union began to offer online banking services to all of its customers [8].In the year 2006 80% of US banks had began to offer some variation of online banking [8]. It was a slow process to move from the tried and true brick and mortar banking that the majority of the public were,if not entirely satisfied with, were familiar with and accustomed to through decades of useage. The move from brick and mortar to click and mortar was not all advantages, there were many issues which had to be addressed such as ease of access, security and the education of the public about phishing scams and various other crimes which can occur by storing information online. Some banks offered advice to the elderly and people who may not know so much about computers advice in an attempt to cut down on online banking crimes I have refereneced an example of one such site targeting the elderly to dispense such advice [9]

2.4 Advantages of Online Banking

2.4.1 Convenience

There are many advantages to online banking which I will discuss in this section the most prominent of these advantages is convenience. Bank customers no longer have to drive or walk to the bank or to an ATM to check their balance and with online payments customers will never have to withdraw money unnecessarily. You can now easily transfer money between accounts online and set up standing payments to pay your bills which is a lifesaver for some people. You can also use your mobile to view your balance and statements anywhere using mobile data and online banking applications, this allows users to access their account from anywhere in the world depending on the availability of reception.

2.4.2 Speed

The speed of which a bank customer can access their account information is now determined by the speed of their internet. Before the advent of online banking you would have to wait in line at an ATM to see your statements or to withdraw money to pay your bills, now that online banking has become virtually ubiquitous in the modern world you no longer have to wait in lines to perform activities and tasks relating to banking.

2.4.3 Competition

The rise of online banking means that banks are constantly trying to out do each other in terms of their banking applications. The customer now has much more choice in choosing banks since the advent of online banking. The advent of online banking has led to some banks maintaining only an online prescence.[10]

2.4.4 Creation of Jobs

The switch from physical banking to digital banking has created numerous jobs in the software industry especially in terms of cyber-security as banks increasingly worry about security braches and hacks. Any hacks that occur are viewed as the fault of the bank and allows many customers to migrate to other banks, that is why many banks spend a lot of money to secure data on their servers which creates jobs and stimulates the economy.

2.5 Disadvantages of Online Banking

2.5.1 Security

Storing details in a digital environment is a lot less secure than storing them in a non digital format. This happens because the data is stored on a server and if phishing scams getting a user's password are successful they could find that their bank account has been drained of money. There are many other forms of attacks which could occur such as keylogging, network monitoring and on public networks your data is far less secure. Although banks have taken many counter measures to prevent such attacks from occuring sometimes they slip through the cracks of cyber-security analysts[11]. Most people find that a little less security is a fair trade-off for the convenience online banking provides.

2.5.2 Layoff of Bank Employees

As the migration of banking from physical to digital takes place it leads to a surplus of banking employees which are made redundant. This occurs because of the ease of access of online banking and this limits the need for more tellers in many banks.

2.5.3 Increased Crime

As banks migrate there is an increase of cyber-criminals who try to gain access to the banks. According to one article 25% of all malware hits financial services [12]. This increase of crime leads to an increased number of compromised accounts on a banks server.

2.6 The Overall Effect of Online Banking

The overall effect of online banking has had a stunning impact on our world. The availability of a bank 24/7 has provided countless benefits to both consumers and businesses alike. The advantages in my mind as in the minds of many others far outweigh the disadvantages of online banking.

Chapter 3

Methodology

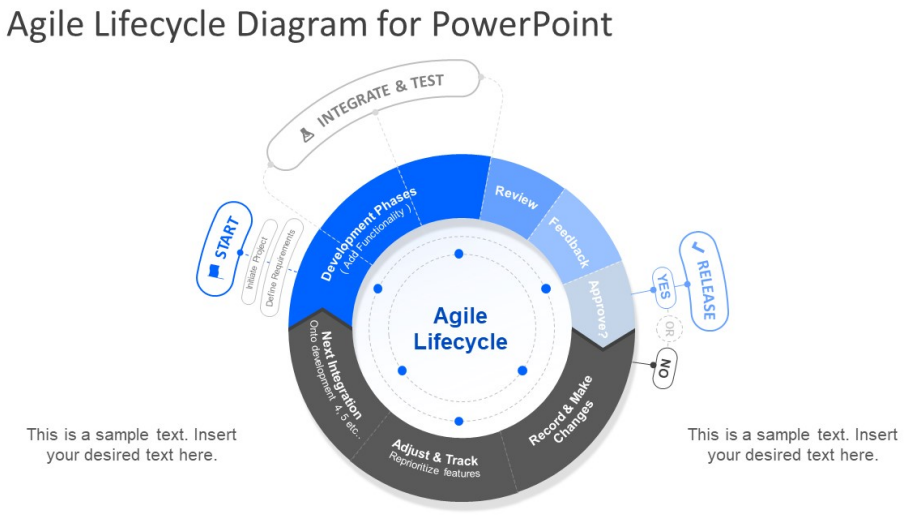
3.1 Overview of Methodology

In this project I utilized the Agile methodology, I used Sprints to coordinate my work and after each sprint I performed module testing and regression testing. I also used Test Driven Development which involves writing tests and writing just enough code to make them pass and then going back and refactoring the code.

3.2 Agile

3.2.1 What is Agile?

Agile is a software development methodology which according to the agile manifesto website [13]: "Individuals and interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan". There are many variations of agile [14] such as Scrum, XP, Kanban and many more. All Agile methodologies follow the same principles outlined in the Agile Manifesto[13] For the purposes of this project I'll be using Kanban. All Agile methodologies stress continuous improvement and incremental delivery.



[15]

3.2.2 Why Kanban?

Kanban [16] allows for a visualized workflow and was also easy to implement into the project. It allows for incremental development and a continuous improvement of features. In contrast to waterfall, Agile methodologies also allow the developer to go back and improve upon features. In summary Kanban offered me a visualized workflow and also allowed me to divide my sprints into daily or weekly tasks which I could complete in a relatively short time.

3.2.3 How I Applied Agile To This Project

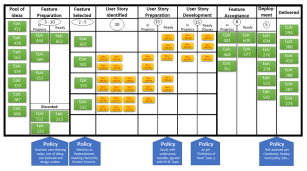


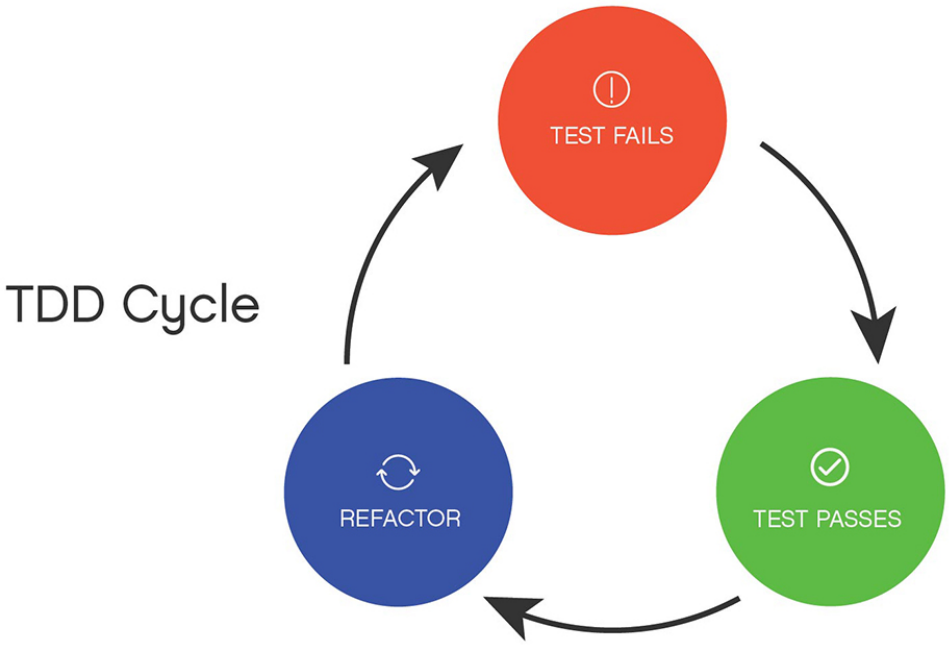
Figure 3.1: Image of Kanban board from wikipedia

During this project I used a Kanban board on Github [17] to coordinate my sprints(Backlog) and added the issues that needed to be finished to the Kanban board. When an issue was added it was automatically moved to the "To Do" section of the Kanban board when it was in progress it was moved to the "in progress" section and finally when it was done I moved it to the "Done" section of the Kanban board. The Kanban board helped create a visualized workflow which helped productivity as I could visualize which issues I would be working on that day and also I could coordinate the Kanban board with my sprints to segregate work I needed to accomplish in a fixed time-frame.

3.3 Test Driven Development (TDD)

3.3.1 What Is Test Driven Development?

Test Driven Development or TDD is a form of development which focuses on writing the tests of a function before coding it and then writing enough code to make sure that it passes the test and then go back and refactor it[18].



[19]

3.3.2 How I Applied Test Driven Development To This Project

I wrote simple tests as I developed this code, for example for the statements section I wrote a test such as "Output only the statements of the user, output must be in form of location, cost, name, date" I then proceeded to write just enough code to make it pass and then went back and refactored it.

3.4 Time Management

3.4.1 How I Handled Multiple Projects

I was able to handle multiple projects at a time by defining sprints in my final year project and assigning myself daily tasks using the Kanban Board,

sometimes I had issues in completing these daily tasks but I always tried to complete them in a timely and effective manner. Sometimes I experienced scope-creep with a task where I had to add more features to the task to make the project viable.

3.4.2 Issues I Faced With Time Management

I faced some issues with time management as the workload was fairly heavy, I had to cut some projects short and leave out some features of various projects to ensure that all projects were completed to an acceptable level.

3.5 Version Control

I used GitHub to manage the versions of this application and to incrementally add code to my repository. I also used GitHub for the Kanban board and to manage any issues that arose during the course of development.

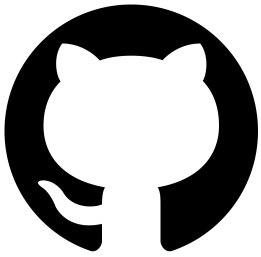


Figure 3.2: Github Logo from github.com

Chapter 4

Technology Review

About seven to ten pages.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XMl, WSDL, JSON, JAXP.
- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

4.1 XML

Here's some nicely formatted XML:

```
<this>
  <looks lookswhat="good">
    Good
  </looks>
</this>
```

Chapter 5

System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

Column 1	Column 2
Rows 2.1	Row 2.2

Table 5.1: A table.

Chapter 6

System Evaluation

As many pages as needed.

- Prove that your software is robust. How? Testing etc.
- Use performance benchmarks (space and time) if algorithmic.
- Measure the outcomes / outputs of your system / software against the objectives from the Introduction.
- Highlight any limitations or opportuni-ties in your approach or technologies used.

Chapter 7

Conclusion

About three pages.

Appendices

Appendix A

Preamble & Intro

- Link to my github

Bibliography

[1] A. Morgan, "The modern application stack – part 1: Introducing the mean stack." https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack.

[2] Educba, "Is mongodb open source?." https://www.educba.com/mongodb-open-source/.

[3] F. Hámori, "How developers use node.js - survey results." https://blog.risingstack.com/node-js-developer-survey-results-2016/.

[4] J. Rocheleau, "Mongodb for beginners: Introduction and installation (part 1/3)." https://www.hongkiat.com/blog/webdev-with-mongodb-part1/.

[5] CodeCondo, "Mean stack vs lamp stack." https://codecondo.com/mean-stack-vs-lamp-stack/.

[6] Dashlane, "The 3 simple reasons why dashlane uses react." https://blog.dashlane.com/the-3-simple-reasons-why-dashlane-uses-react/.

[7] W. various, "Node.js." https://en.wikipedia.org/wiki/Node.js.

[8] R. Sarreal, "History of online banking: How internet banking went mainstream." https://www.gobankingrates.com/banking/banks/history-online-banking/.

[9] B. O. Ireland, "Help for older people." https://www.bankofireland.com/security-zone/help-for-older-people/.

[10] J. Pilcher, "25 digital-only banks to watch." https://thefinancialbrand.com/69560/25-direct-online-digital-banks/.

[11] G. Blog, "Cyber bank robberies contribute to $1 trillion in cybercrime losses." https://www.globalsign.com/en/blog/cyber-bank-robberies-contribute-to-1-trillion-in-cybercrime-losses/.

[12] Z. Doffman, "Cybercrime: 25% of all malware targets financial services, credit card fraud up 200%." https://www.forbes.com/sites/zakdoffman/2019/04/29/new-cyber-report-25-of-all-malware-hits-financial-services-card-fraud-up-20#44d8cd4b7a47s.

[13] various, "Manifesto for agile software development." https://agilemanifesto.org/.

[14] S. K. Magdalena Maneva, Natasa Koceska, "Measuring agility in agile methodologies." https://core.ac.uk/reader/149219742.

[15] SlideModel, "Agile process lifecycle diagram for powerpoint." https://slidemodel.com/templates/agile-process-lifecycle-diagram-powerpoint/.

[16] atlassian, "Kanban how the kanban methodology applies to software development." https://www.atlassian.com/agile/kanban.

[17] U. Kearns, "Kanban board on github." https://github.com/Ultan-Kearns/AppliedProject/projects/3.

[18] W. Aejmelaeus, "Test-driven development." https://core.ac.uk/reader/37986089.

[19] M. Warcholinski, "Test-driven development (tdd) – quick guide." https://brainhub.eu/blog/test-driven-development-tdd/.