BuyIT
An E-Commerce Web Application

Final Year Project
B.Sc.(Hons) in Software Development

BY
RICHARD COOKE

MAY 7, 2020

Advised by Dr. John Healy & Mr. Gerard Harrison
DEPARTMENT OF COMPUTER SCIENCE AND APPLIED PHYSICS
GALWAY-MAYO INSTITUTE OF TECHNOLOGY (GMIT)

Contents

List of Figures

Abstract:

E-commerce continues to play an ever expanding role in modern society, multinational corporations and the information technology sector. E-commerce is now the leading way in which business to consumer transactions occur. This document details the creation of a modern-day e-commerce web application. The document details how this was planned, solved and the methods that were used in order to meet the requirements of a modern day customer. The solution to this was to create a three-tier web application that allows user to view products, add them to cart, review them, order them and personalize user details. Restrictions on the creation and deletion of products are limited to those with administrative rights.

Author:

This project was developed as part of the 15 credit module, Applied Project and Minor Dissertation, by Richard Cooke, final year student of the Computing and Software Development course at Galway-Mayo Institute of Technology(GMIT).

Acknowledgements:

The author would like to thank project supervisor Mr. Gerard Harrison and module supervisor Dr. John Healy for their time and support throughout the project development.

Chapter 1

Introduction

This chapter serves as an introduction to the project. In it, the various aspects of the project are discussed and the main objectives of the project. Each of the chapters are briefly discussed as well, detailing what each contains and what is discussed in each.

Throughout the first three years of study at GMIT, students, myself included, were educated on a broad range of technologies in both hardware and software. This was conceived so that graduating students would be better prepared for a wide range of challenges in the post-graduate workplace.
At the commencement of our 4th year, we were informed that a year-long, 15 credit project would have to submitted by the end of the second semester of the current academic year. Following this it was determined that I would perform this task as part of a 3-person group, however this group would later split with both parties keeping the original idea in slightly different formats.
With input from the project supervisor, I began to research which technologies could form the basis of this project. Discussions with the project supervisor occurred on a weekly basis to ascertain that the project was remaining within the scope of what would be expected of a level 8 project. Due to the overwhelming presence of e-commerce in modern day life I found this to be a suitable area to explore project wise, I also determined this would be an accurate way to gauge the growth in my software development skills as the end of year project for our first year at GMIT had a similar basis.

1.1 Project Objectives

The main goal of this project is to produce a web application that mimics a users experience on e-commerce sites such as amazon. Along with this the author would like to gain a better understanding of e-commerce, the web-technologies involved in e-commerce development and to attempt to gauge the developers own development in working with web technologies.
The project can be divided into 2 segments, the applied project itself and this dissertation. The dissertation will be used as a means for containing the research conducted in the projects development whereas the applied project will outline the technologies involved in the projects development.

The objectives of the applied project where as follows :

Produce a simple, easy to use and understand, web application. This application will make use of a number of complex algorithms with a sophisticated back end server and database. The complexities of these features are to remain hidden from users and users must be presented with an easily understandable and navigable front-end, which should not require the user to have anything above base understandings of web applications.

Gain further understanding of web technologies. The development of this project will be used as a time for the author to both learn and work with new web technologies and for the author to further develop their understanding of other, previously used, web technologies. This will be done to try and maximize the learning outcomes of the module.

Create an e-commerce web application that meets modern expectations. The end goal of the applied project is to have produced a web application that has an environment similar to that of modern, multinational e-commerce web applications such as Amazon or eBay.

Complete the given assignment as a lone venture. This project will be undertaken as solo endeavour. This is to be achieved by adhering to industry standards and methodologies, which has been done to simulate the expected experience of working on a similar scenario in a real workplace scenario. As the entirety of development responsibilities lay solely with a singular developer it is imperative that the project is approached so that maximum efficiency is aimed for.

The objectives of the dissertation where as follows:

Introduce the concept of the project. The author of this paper will aim to provide the user with an introduction to the project that accurately describes its inspirations and end-goals.

Provide an understanding of web technologies. The number of web technologies available for web applications continues to grow at a near exponential rate. Within this paper the author will discuss a range of these technologies, The author will also argue why the technologies that were used were the most appropriate for the projects development.

Outline the development of the applied project. One of the main aims of the paper is to give its readers a complete breakdown of the projects development cycle from the initial conception, to the research involved and on to the eventual completion of the applied project. The methodologies and technologies used in the projects development, along with an evaluation of the system as well as a discussion of the systems design. Following this, the issues that the were encountered by the author throughout the development of the project in both the applied project and in other scenarios, how these problems were solved and what would have been done differently if the project were restarted. To conclude this paper the author will evaluate the project as a whole.

1.2 Metrics for Success and Failure

In an attempt to control and more easily manage the development of the project, the author outlined the metrics that would mean either success or failure for the project, in its early stages. Doing this allowed for the author to track their progress and to ascertain that the project maintained its course in achieving the project objectives that were earlier outlined.

The metrics for which the success of the applied project and the dissertation are as follows:

A comprehensive dissertation that can be understood by anyone, regardless of their initial knowledge level of the technologies implemented in the applied project. This was achieved by making conceptual drafts for each section that was to be included in the finalized version of the dissertation and these early drafts were shared with and read by students who were attempting similar projects, this service was also done for these students who wanted their drafts read in an attempt to cultivate better understanding of what expectations were for what was being demanded.

An easily used web application, that fully functions. This was achieved using similar methodology as with the dissertation. As the application was being developed, beta versions where distributed to fellow students of the author so as to ascertain that the application was meeting the preassigned objectives of being easily used and accessed and maintaining full functionality.

1.3 Chapter Outline

This paper has been organized into a standard layout of chapters, with the purpose of each of these chapters being to explore the various, different, aspects of the projects. These chapters are outlined as follows:

1.3.1 Methodology

This chapter will outline the approaches that the author used in the planning, organization, management and development of the project. The section aims to enlighten the reader on how the projects progressed from research to the final applied project. The author will discuss the what methodologies where implemented in the development of the project and why these methodologies were chosen.

1.3.2 Technology Review

This chapter will outline the technologies that were used in the final version of the project. The technologies will be explained by the author and their implementation in the project will be outlined along with why they were chosen as the technologies to be used in the projects development over other, similar, technologies.

1.3.3 System Design

This chapter will outline the architecture and design of the web applications system. Using code snippets and visual diagrams to help the reader attain a basic understanding of the web application design. This chapter will provided an in-depth exploration of all the layers of the system, the data, logic and presentation layers.

1.3.4 System Evaluation

This chapter will provide an evaluation of the software developed for the final version of the applied project. The author will evaluate areas of the system such as its robustness, scalability and testing of the system. The results of this evaluation will be used as a measurement to compare with the objectives outlined in the introduction section. The author will also discuss what opportunities could have been taken to improve the system and software developed.

1.3.5 Conclusion

This chapter will provide a review of the goals outlined at the onset of the project. It will highlight any findings from the System Evaluation chapter, giving a final analysis on what discoveries were made through the development of the project, before the author starts a brief discussion on their experience of the project.

1.4 User Requirements

- User must be able to register an account
- User must be able to log in with email and password
- User must be able to log in with Google account
- Log in must be persistent throughout a session
- User must be able to log out
- User must be able to add personal details
- User must be able to view product
- User must be able to add product to cart
- User must be able to view cart
- User must be able to remove cart item
- User must be able to review product
- User must not be able to access "Admin" section
- Admin must be able to view list of user
- Admin must be able to add products
- Admin must be able to edit products
- Admin must be able to delete products
- Admin must be able to attach picture to product

1.5 Github

In the case of this project was used for version control. The git repository also contains a readme which contains instructions on how to install and run the project on a local machine.
The Github repository can be found at:
https://github.com/CookeRichard94/fyp

Chapter 2

Methodology

In this chapter the author will discuss the methodologies used to approach the project in areas such as planning, organization, development and management of the project. This section will aim to provide the reader with apt insight into how the project developed from its research stage into the final product that it is now.
As part of the 4 year course of Computing in Software Development a number of modules, such as Systems Analysis and Project Management, were designed to enlighten students on the different approaches or methodologies that could be taken throughout a projects development. Frequent methodologies which were taught as part of these modules were extreme programming, waterfall and RAD (Rapid Application Development).

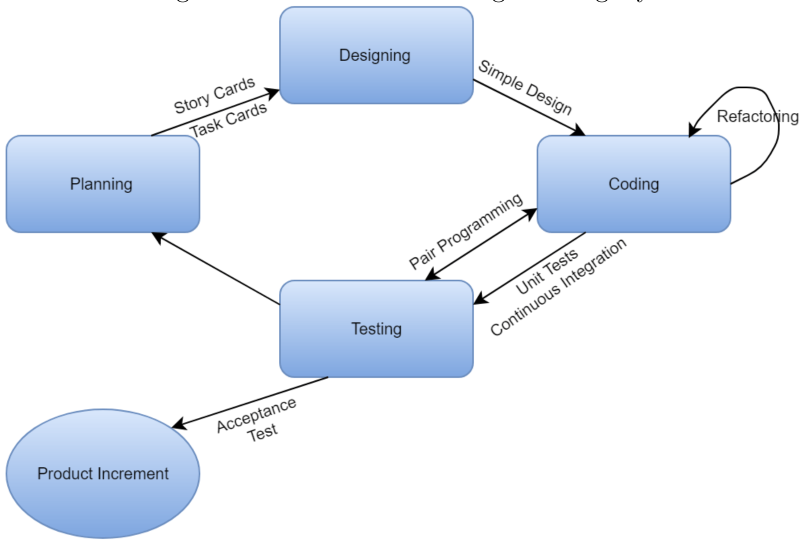
2.1 Research Methodology

In the case of this project, both Qualitative Research and Quantitative Research were used as the approach to conducting the appropriate research. Qualitative research is an exploratory research method used to gain a greater understanding on a topic, it is also used as an aid to further quantitative research. The qualitative research was carried out by closely examining similar web based technologies to the the end goal of the project, such as Amazon, eBay and many other e-commerce based applications. It was at this point that the approach to the project in aspects such as software development methodologies were determined.
As the qualitative research continued a shift towards more quantitative research occurred, throughout this research a better idea of what features were commonly expected and found on web applications similar to that of the project.

2.2 Software Development Methodology

The software development methodology appropriated throughout the development of this project was Extreme Programming[7]. Extreme Programming is considered to be a highly agile software development methodology that aims to produce high quality software. This methodology is noted for allowing the the development to change and adapt to the request of a client or customer. The use of this methodology in the projects development cycle was important as it, like all agile methodologies, allowed for any changes to have minimal impact on the project.
Shortly into the development cycle it became clear that Extreme Programming was highly appropriate in this project scenario as it allowed for the adaption to changing requirements on the advice of the project supervisor as part of the weekly stand-up meetings. As part of extreme programming features deemed to be priorities are ear-marked to be completed first, the test-driven nature of extreme programming allowed for confirmation that these features were in working order, which in turn allowed for the focus to be moved along down the priority list.
Further reasoning as to why this methodology was chosen was because of the amount of potential new technologies involved with the development of this project. As the author may not have had previous experience in working with some of the technologies, the full libraries and capabilities of these technologies would not be known to the author and as such the agile methodology allowed for the author to implement any features and knowledge developed throughout the project and keep the impact of any changes to a minimum.

Figure 2.1: The Extreme Programming Cycle



2.2.1 Weekly Meetings

As part of the project it was determined that participants would engage in weekly stand-up meetings with a preassigned project supervisor. In this scenario the supervisor acts as a pseudo-client, which allowed for better implementation of an agile software development methodology. The initial meetings centered on updates on a decision of a project subject, this is where the majority of the qualitative research was conducted, and whether or not this subject would be within the scope of the project statement. After the project subject had been determined the stand-up meetings mainly consisted of providing the supervisor with updates on the development process, with feedback on said updates and a discussion on what next would be implemented as part of the project. The weekly stand-up meetings can also be said to bear a strong resemblance to the stand-up meetings integrated as part of the SCRUM development methodology

2.3 Development Tools

In the process of developing this project two main development tools were used, Github and Visual Studio Code. Github is a web-based technology that allows for hosting of software via the command line tool git. Git is an open-source version control system that allows for changes to the project to be pushed and stored on github. In the case of this project github was used as the version control system as it was most familiar to the author and the tracking of progress was outlined as an important aspect of the project.
Visual studio code is a source code editor. In the case of this project visual studio code was chosen as it supports multiple file types like those present in an Angular based project and the integrated terminal allows for the integration of libraries that were needed throughout the projects development.

Chapter 3

Technology Review

This chapter will focus on the technical aspects of the projects by reviewing the technologies that make up the final version of the applied project. It will outline the reason these technologies were chosen, what the advantages and disadvantages of using these technologies was and the role of these technologies within the system.

3.1 Overview

For this project it was determined that a full stack development would be preferred, with this in mind it was important to determine what technologies would be suitable for a project such as this and which of these technologies would be most compatible with one another. Prior to taking on this project, the author had had experience using the MEAN stack in one previous module, the MEAN stack is an acronym for the use of the technologies MongoDB, Express, Angular and NodeJs. As previously mentioned, the author wished to use this project to both learn of new technologies and to polish up on some previously exercised technologies. For this purpose the stack used in the development of this project was as follows:

- Angular for front end purposes (Previous experience)
- NodeJs for server-side purposes (Previous experience)
- Firebase for authentication purposes (New Technology)
- Firebase for database purposes (New Technology)
- Firebase for application deployment (New Technology)

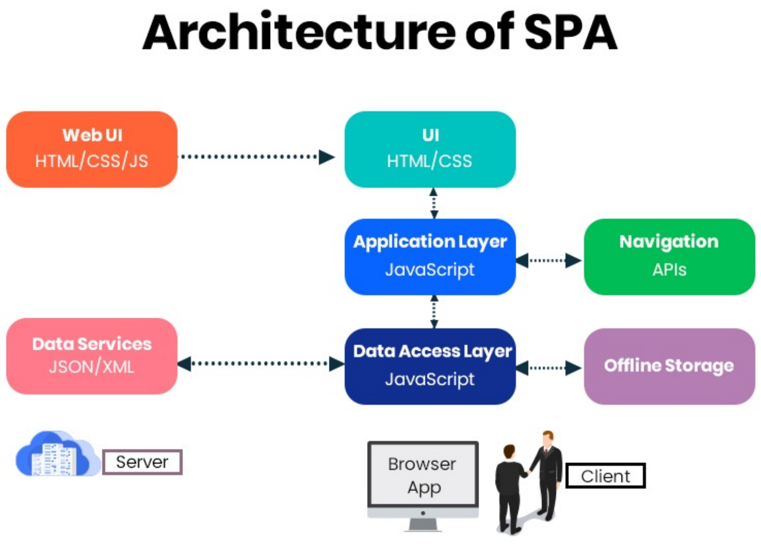
3.2 Core Technologies

This section will discuss the core technologies that were implemented in the development of the web application.

3.2.1 Angular

Angular is a typescript based, web-application framework, used to develop single-page web-applications. A single-page web-application is one which has all of the functionality of a multi-paged web-application but without the need to refresh the browser whenever the page of the web-application has been changed. Other frameworks that can be used to create single-paged web-applications are React, Django and Ionic. The popularity of single-page web-application has continued to grow in recent years. This rise in popularity can be linked back to the free flowing feeling these applications provide to the user rather than the stop-start mechanisms of a multi-page based web-application [14]

Figure 3.1: The Architecture of Single-Page Application



Angular was first released on October 20th 2010, as AngularJs, before experiencing a complete re-write and becoming the entity that we currently know it as today. As of the time of the writing of this paper Angular is currently on version 9, which was released on February the 6th 2020 and continues to grow the framework while fixing bugs [4].

For the purpose of this function Angular was chosen by the author because of the authors previous experience with it and being able to quickly discern that it was a reliable framework that was cooperative with other components. The author felt that improving on previous learning of the framework through independent learning would be beneficial as although the previous experience had been useful it resolved around MEAN stack implementation only whereas this proposed project would move away from that approach.

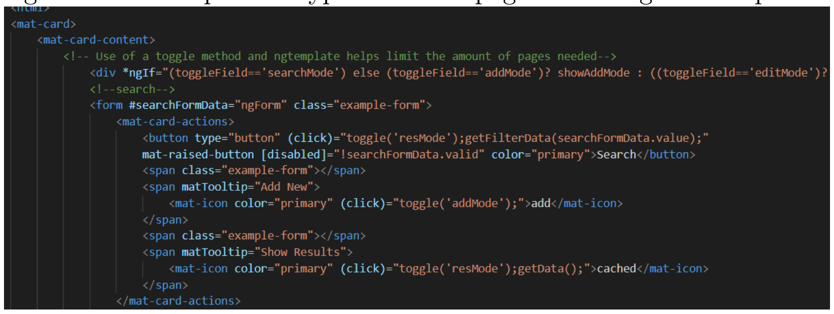


Figure 3.2: Example of a typical HTML page in an Angular component

```
<mat-card>
    <mat-card-content>
        <!-- Use of a toggle method and ngtemplate helps limit the amount of pages needed-->
            <div *ngIf="(toggleField=='searchMode') else (toggleField=='addMode')? showAddMode : ((toggleField=='editMode')?
            <!--search-->
            <form #searchFormData="ngForm" class="example-form">
                <mat-card-actions>
                    <button type="button" (click)="toggle('resMode');getFilterData(searchFormData.value);"
                    mat-raised-button [disabled]="!searchFormData.valid" color="primary">Search</button>
                    <span class="example-form"></span>
                    <span matTooltip="Add New">
                        <mat-icon color="primary" (click)="toggle('addMode');">add</mat-icon>
                    </span>
                    <span class="example-form"></span>
                    <span matTooltip="Show Results">
                        <mat-icon color="primary" (click)="toggle('resMode');getData();">cached</mat-icon>
                    </span>
                </mat-card-actions>
```

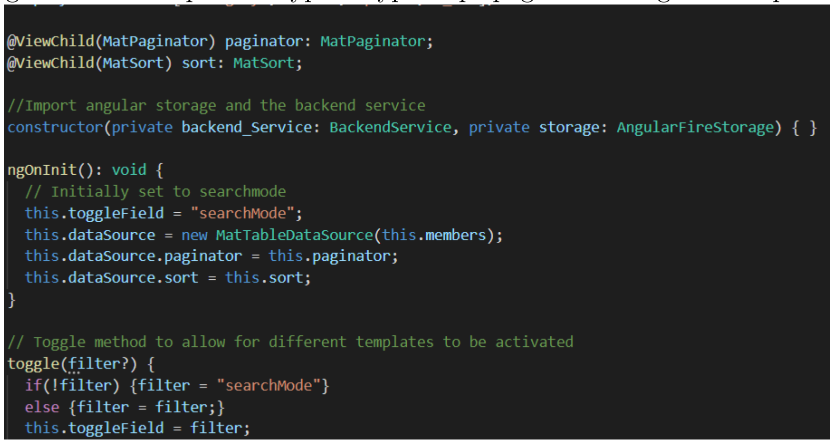


Figure 3.3: Example of a typical typescript page in an Angular component

```
@ViewChild(MatPaginator) paginator: MatPaginator;
@ViewChild(MatSort) sort: MatSort;

//Import angular storage and the backend service
constructor(private backend_Service: BackendService, private storage: AngularFireStorage) { }

ngOnInit(): void {
  // Initially set to searchmode
  this.toggleField = "searchMode";
  this.dataSource = new MatTableDataSource(this.members);
  this.dataSource.paginator = this.paginator;
  this.dataSource.sort = this.sort;
}

// Toggle method to allow for different templates to be activated
toggle(filter?) {
  if(!filter) {filter = "searchMode"}
  else {filter = filter;}
  this.toggleField = filter;
```

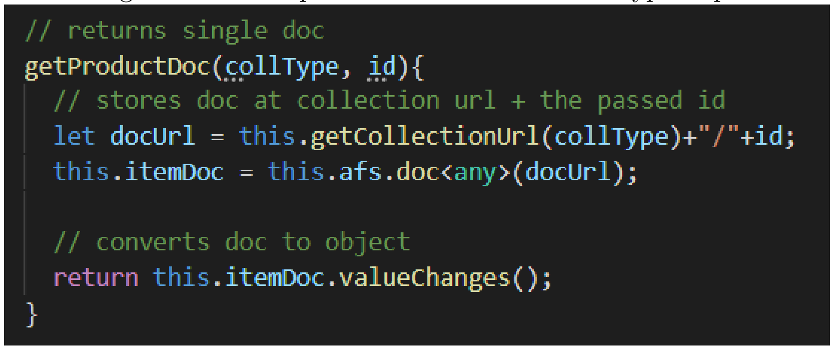
Angular-CLI

Angular-CLI is a command line tool that used to aid in the development of an Angular based project. Through the Angular-CLI the application can be built and served and new services can be easily generated[6].

Typescript

Typescript is a superset of the programming language Javascript.It is compiled as javascript, which essentially makes it javascript plus additional features. Developers with experience of javascript will easily be able to adapt to using typescript when developing a web-application[15].

Figure 3.4: Example of a function written in typescript



```
// returns single doc
getProductDoc(collType, id){
  // stores doc at collection url + the passed id
  let docUrl = this.getCollectionUrl(collType)+"/"+id;
  this.itemDoc = this.afs.doc<any>(docUrl);

  // converts doc to object
  return this.itemDoc.valueChanges();
}
```

HTML

HTML stands for Hyper Text Markup Language. HTML constitutes the "view" that the user of a web-application has. It is used to provide structure to the web-page and the features it creates on the web-page are provided functionality by the javascript/typescript.

CSS

CSS stands for Cascading Style Sheets. CSS plays no role in the functionality of the web-application. The purpose of CSS is to make the web-application look appealing to users which is done by promoting a consistent styling throughout the entire web-application.
There are both advantages and disadvantages to using Angular framework: [5]

Advantages of Angular

- Model-View-Controller architecture of Angular allows the logic side of the application to be separate from the UI side of the application. Thanks to this architecture, problems within an Angular application can be more easily isolated.
- Custom directives, Angular allows for custom directives such as ngModel, which provides two way data binding to HTML form elements, directives such as these allow for improved HTML functionality and for improved and more dynamic client-side applications.

Disadvantages of Angular

- Lack of CLI documentation. Developers have continually expressed concern over the lack of information in Angulars CLI documentation. Though the command line is highly usable within an angular project, a developer is likely to have to source information on potential commands from non-official sources.

- Steep Learning Curve. Many developers who are familiar with javascript and have branched into javascript based frameworks have made note of the fact that Angular, compared to other frameworks such as ReactJs, features a much steeper learning curve. This steep learning curve is due to the vast of array of topics and aspects of the framework that can be covered.

3.2.2 NodeJs

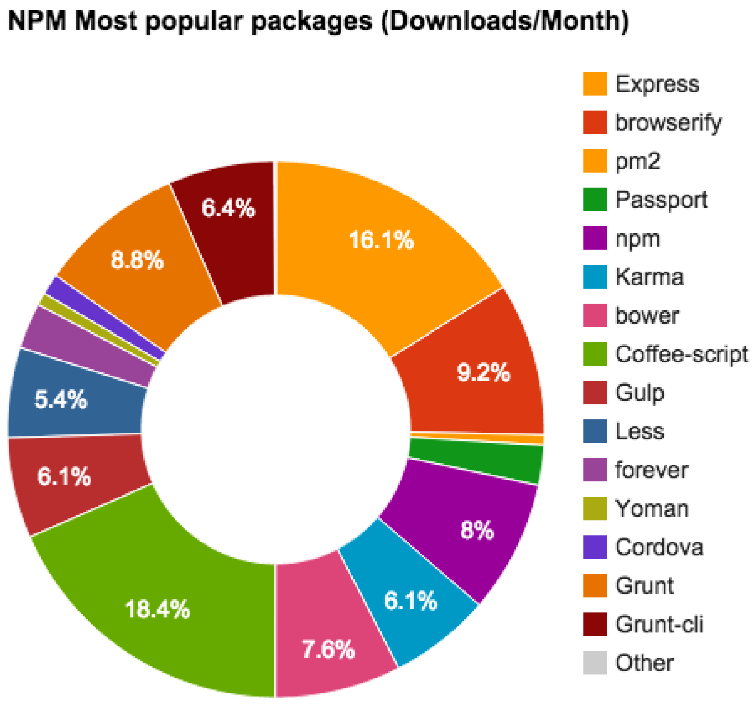
NodeJs is a cross-platform javascript run-time environment that allows for javascript to run outside of a browser. NodeJs features built in HTTP functionality, this allows a developer to utilize this HTTP functionality in a local environment wherein usually this would not be possible [1]. In the case of this project NodeJs is used for server-side purposes, meaning that the server-side and the front-end of the project both function using javascript which allows for a minimization of needed maintenance of the components of the project.
The reason NodeJs was used as a par of this project was that the author had previous experience with the technology and because of it's known compatibility with the Angular framework. Due to this easily maintained relationship with Angular, this offered more time to be attributed to exploring the new technologies being introduced to the author. Although the author has previous working experience with NodeJs, the environment on this occasion did not have NodeJs working in conjunction with Express as had been previously done.

Node Modules

NodeJs allows developers to make use of extra packages called Node Modules or NPMs. These packages usually provide extra functionality to an application, these packages provide a range of different functionality such as streamlined server connection and user authentication [12]. Throughout the development of this project NPM was used for providing server side connection to firebase and for the control of Angular modules and materials.

There are both advantages and disadvantages to using NodeJs[3]:

Figure 3.5: Popular NPM packages



Javascript

Javascript is a programming language responsible for the functionality of web-pages. The javascript of a web-page allows for communication with the sever side of the application. In the case of this project, and many Angular based projects, typescript is used, which is a subset of javascript[11].

Advatanges of NodeJs

- Easily Scalable. NodeJs can be used to scale an application either vertically or horizontally if needed. Single nodes within the server can make use of different resources which is not always available on javascript servers.

- Single Programming Language. Javascript is currently renowned as the most popular programming language in the world, as such, the fact that NodeJs can be completely run in javascript makes it a highly accessible server. The single use of javascript also makes NodeJs highly compatible with most web-application frameworks as the front-end functionality of these application is most commonly written using javascript as these applications are built to run on web browsers and most web browsers support javascript.

Disadvantages of NodeJs

- Unstable API. The API of NodeJs is consistently changing and with this consistent change is a lack of stability. These new API changes often do not offer backwards compatibility and as such will require regular maintenance to occur if the application is to be stable and retain all functionality that was intended.

- Library Support. Compared to other programming languages such as Python and its flask server, javascript lacks a robust library system. This means that in some cases like in some parsing operations that NodeJs will have to rely on common libraries to complete these operations. As such it can go without saying that because NodeJs is reliant on javascript it actually adopts all of the weaknesses of javascript.

3.2.3 Firebase

Within this project Firebase was used as both a database and for user authentication. Firebase is a Backend-as-a-Service (BaaS) created in 2011 before later being acquired by Google and being developed into a cloud platform. In the case of database use Firebase is largely similar to other document based databases such as MongoDB, this allowed for the learning curve to be easily navigated with any prior experience of working with document based databases to be of great help. However unlike traditional databases, Firebase removes the need for a back end server as would be usually expected on some stack development such as on the MEAN stack. Client SDKs allow for direct interaction to back end services which eliminates the need for middle-ware in these applications [10].

Firebase was also used for user authentication purposes throughout the project. Firebase provides many SDKs and back end services that allow for user verification in a multitude of ways. Users can opt for sign in through social platforms such as Facebook or Twitter, standard E-mail and Password registration or through a Google(G-mail) account. In the case of this project the author has utilized the Google and E-mail/Password authentication [8]. When a user has been authenticated they will be granted an access token which will be attached to a unique id and either their email or the id of their social platform

Figure 3.6: Traditional Infrastructure Vs Firebase

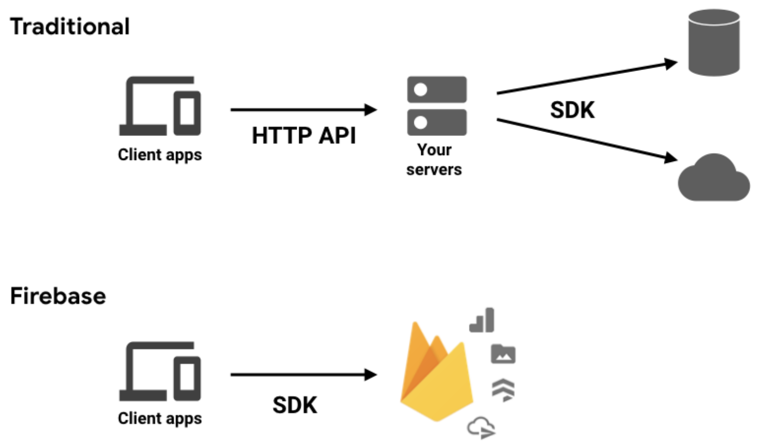
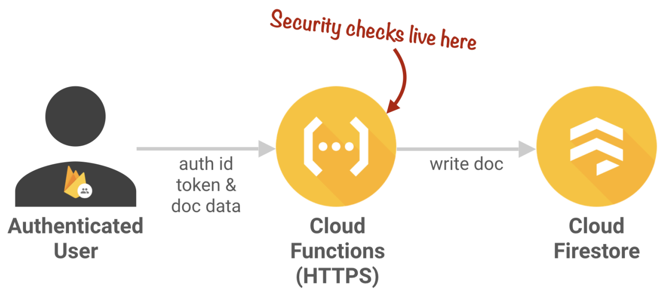


Figure 3.7: How Firebase Authentication Works



Firebase Deployment

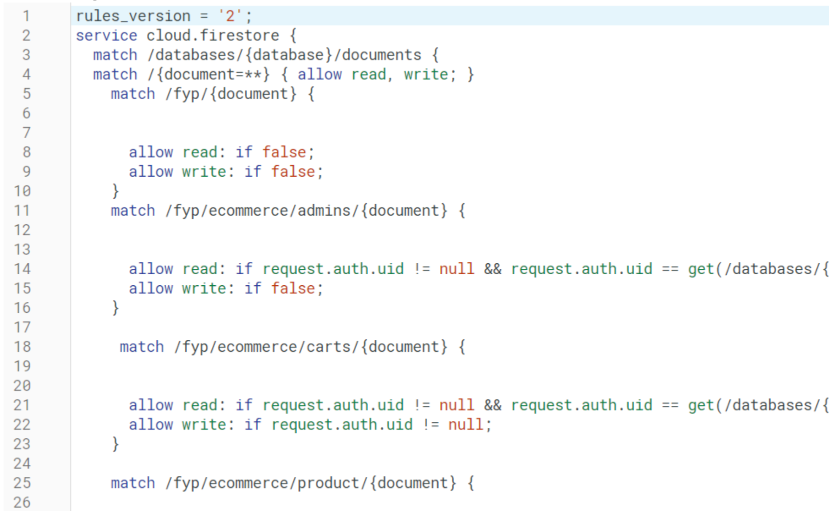
Firebase also provides a variety of other products such as hosting of applications, in the case of this project firebase deployment was used for a number of reasons, the first of which was that because of the uniqueness of the Firebase connections were no back end server needs to be established this limited the use of some other deployment platforms such as Heroku, I found this to be a considerable disadvantage to using a Firebase constructed database. Aside from this, the deployment of an application to Firebase cloud servers was found to be a much simpler process than other deployment methods. In the case of Firebase, all that is required is to login to the developers Firebase account, use the command ”firebase init” in the command line, check off a series of options and then to use the command ”firebase deploy” [9] where upon the developer is presented with the URL which their application is now being hosted on. All in all the deployment process for this application, where Firebase was concerned was less than 5 minutes.

A fully deployed version of the web-application is available on: https://fyp-server-75156.web.app/

Security

Although a portion of the security related to Firebase was outlined when discussing Firebase user authentication, Firebase provides additional security in its database environment. Within Firebase the developer of a web-application using the database service can establish a set of ”rules” to determine which users, if any, can access, read, write, edit or delete data.

Figure 3.8: Example of a set of Rules in a Firebase database

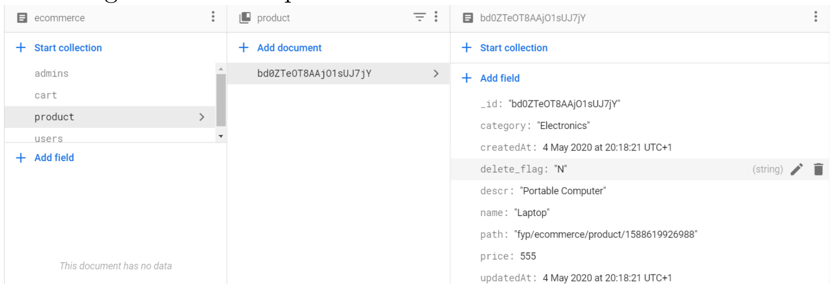


```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
  match /{document=**} { allow read, write; }
    match /fyp/{document} {


      allow read: if false;
      allow write: if false;
    }
    match /fyp/ecommerce/admins/{document} {


      allow read: if request.auth.uid != null && request.auth.uid == get(/databases/{
      allow write: if false;
    }

     match /fyp/ecommerce/carts/{document} {


      allow read: if request.auth.uid != null && request.auth.uid == get(/databases/{
      allow write: if request.auth.uid != null;
    }

    match /fyp/ecommerce/product/{document} {

```

JSON

JSON, or Javascript-Object-Notation, is a lightweight format for storing and transporting data. JSON is text-based, making it readable by both human and machine. JSON is used by Firebase to store data for a number of reasons, it is easy for humans to read and write, it is language independent meaning it can be accessed by a number of programming languages and it supports many different data types[13].

Figure 3.9: Example of JSON within a Firebase Database



```
admins
cart
product
users

bd0ZTeOT8AAjO1sUJ7jY

_id: "bd0ZTeOT8AAjO1sUJ7jY"
category: "Electronics"
createdAt: 4 May 2020 at 20:18:21 UTC+1
delete_flag: "N"
descr: "Portable Computer"
name: "Laptop"
path: "fyp/ecommerce/product/1588619926988"
price: 555
updatedAt: 4 May 2020 at 20:18:21 UTC+1
```

This document has no data

There are Advantages and Disadvantages to using Firebase[2]:

Advantages of Firebase

- Free start. Firebase provides a free starting point for all users, this means that until a certain data amount is reached or a specific service is required the use of Firebase is free to the user.

- Documentation. Firebase provides users with extensive, detailed documentation on the methods, SDKs and the API. This allows for a low-barrier of entry for those who are new to Firebase as the documentation makes it easy to adapt to.

Disadvantages of Firebase

- iOS Support. As Firebase is developed and maintained by Google, it is aimed more at working in conjunction with Android based applications. There are available supports to allow for iOS integration, however in that case it is easier to change to one of Firebases' competitors such as MongoDB.

- Database Limitations. Due to being entirely in JSON, the real time database used by Firebase is limited to simple read, write, update and delete queries.

3.2.4 Bootstrap

Bootstrap is a library of css files available as open-source. Bootstrap is used to aid in the development of the UI of a web-application. Bootstrap is imported as part of the main css file available in every Angular project. As part of the authors project bootstrap is used in collaboration with Angular Materials to create the header file present throughout the entire web-application.

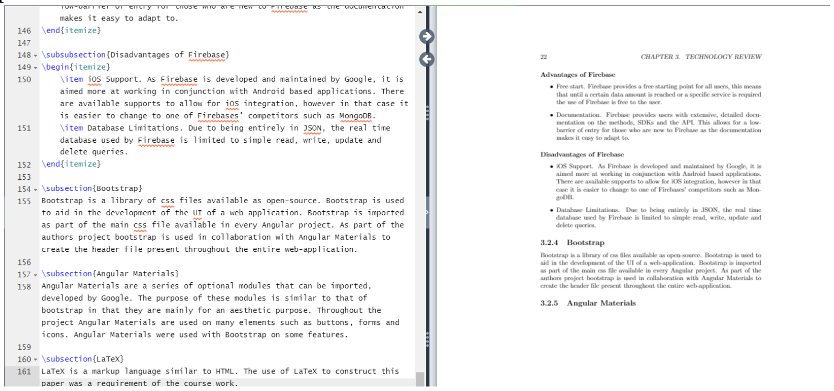
3.2.5 Angular Materials

Angular Materials are a series of optional modules that can be imported, developed by Google. The purpose of these modules is similar to that of bootstrap in that they are mainly for an aesthetic purpose. Throughout the project Angular Materials are used on many elements such as buttons, forms and icons. Angular Materials were used with Bootstrap on some features.

3.2.6 LaTeX

LaTeX is a markup language similar to HTML. The use of LaTeX to construct this paper was a requirement of the course work. LaTeX uses tags to indicate what should happen to the write up when it is compiled. This LaTeX document was constructed and compiled using Overleaf.com. After it has been compiled to a text document, Overleaf allows for the document to be exported as a PDF.

Figure 3.10: Example of Overleaf side by side source code with text document output



```
    low-barrier of entry for those who are new to Firebase as the documentation
    makes it easy to adapt to.
\end{itemize}

\subsubsection{Disadvantages of Firebase}
\begin{itemize}
    \item iOS Support. As Firebase is developed and maintained by Google, it is
    aimed more at working in conjunction with Android based applications. There
    are available supports to allow for iOS integration, however in that case it
    is easier to change to one of Firebases' competitors such as MongoDB.
    \item Database Limitations. Due to being entirely in JSON, the real time
    database used by Firebase is limited to simple read, write, update and
    delete queries.
\end{itemize}

\subsection{Bootstrap}
Bootstrap is a library of css files available as open-source. Bootstrap is used
to aid in the development of the UI of a web-application. Bootstrap is imported
as part of the main css file available in every Angular project. As part of the
authors project bootstrap is used in collaboration with Angular Materials to
create the header file present throughout the entire web-application.

\subsection{Angular Materials}
Angular Materials are a series of optional modules that can be imported,
developed by Google. The purpose of these modules is similar to that of
bootstrap in that they are mainly for an aesthetic purpose. Throughout the
project Angular Materials are used on many elements such as buttons, forms and
icons. Angular Materials were used with Bootstrap on some features.

\subsection{LaTeX}
LaTeX is a markup language similar to HTML. The use of LaTeX to construct this
paper was a requirement of the course work.
```

22 CHAPTER 3. TECHNOLOGY REVIEW

Advantages of Firebase

- Free start. Firebase provides a free starting point for all users, this means that until a certain data amount is reached or a specific service is required the use of Firebase is free to the user.
- Documentation. Firebase provides users with extensive, detailed documentation on the methods, SDKs and the API. This allows for a low-barrier of entry for those who are new to Firebase as the documentation makes it easy to adapt to.

Disadvantages of Firebase

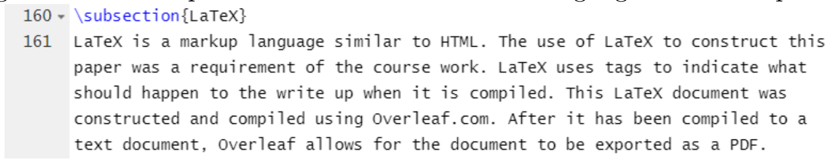
- iOS Support. As Firebase is developed and maintained by Google, it is aimed more at working in conjunction with Android based applications. There are available supports to allow for iOS integration, however in that case it is easier to change to one of Firebases' competitors such as MongoDB.
- Database Limitations. Due to being entirely in JSON, the real time database used by Firebase is limited to simple read, write, update and delete queries.

3.2.4 Bootstrap

Bootstrap is a library of css files available as open-source. Bootstrap is used to aid in the development of the UI of a web-application. Bootstrap is imported as part of the main css file available in every Angular project. As part of the authors project bootstrap is used in collaboration with Angular Materials to create the header file present throughout the entire web-application.

3.2.5 Angular Materials

Figure 3.11: Example of this subsection in latex language before compilation



```
\subsection{LaTeX}
LaTeX is a markup language similar to HTML. The use of LaTeX to construct this
paper was a requirement of the course work. LaTeX uses tags to indicate what
should happen to the write up when it is compiled. This LaTeX document was
constructed and compiled using Overleaf.com. After it has been compiled to a
text document, Overleaf allows for the document to be exported as a PDF.
```

3.2.7 Heroku

Heroku is a Platform as a Service (PaaS), used for the deployment of web-applications to a cloud platform. In the case of this project Heroku was originally intended to be used for the deployment of the web-application however during the research phase on Heroku it was discovered that a Heroku deployed application would need a back end serve meaning Heroku could not be used in conjunction with firebase as firebase eliminates the need for the presence of a backend server.

Chapter 4

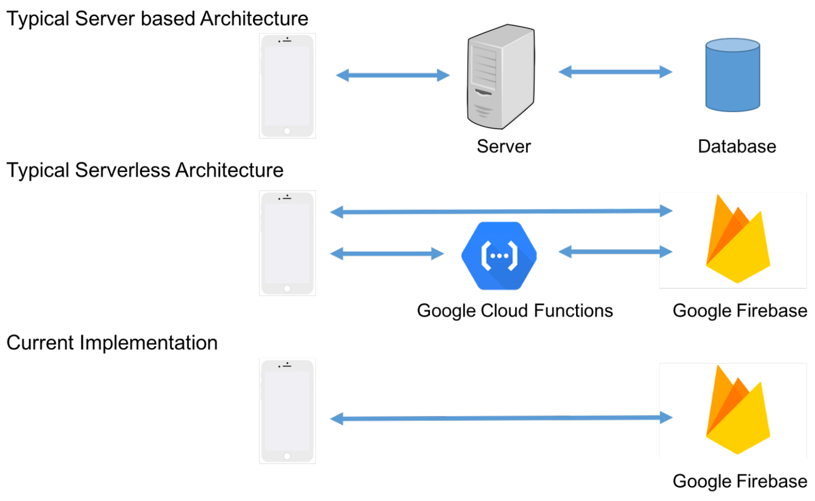
System Design

In this chapter we will discuss the design and architecture of the web-applications system and how the aforementioned technologies are integrated throughout the web-application. Typically in the case of a web-application such as the one defined within this paper, three-tier architecture would be employed as a means to operate the system. However, because Firebase is a backend-as-a-service, technically that would making the architecture of this project 2-tier. However, the two-tier architecture of a Firebase web-application such as this uses pseudo servers, by using imports of angularfire modules, giving the same core functionality as an application with 3-tier architecture.

4.1 Overview

To describe this application in terms of a 3-tier architecture application, the Angular side of operations would wholly conclude the presentation layer, both Angular and Firebase are used to create the "Logic" layer which is conducted by using packages available from the NPM of NodeJs and the Data layer is solely controlled using Firebase technologies. The Angular component passes a request to the backendservice.ts file, which using both Angular and Firebase imports connects with the database and process the request before passing the new data back to the backendservice.ts file to pass on the relevant Angular component.

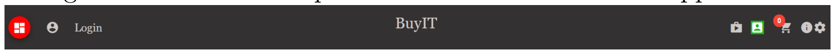
Figure 4.1: Architecture of a Firebase system compared to typical 3-tier architecture



4.1.1 Web Application

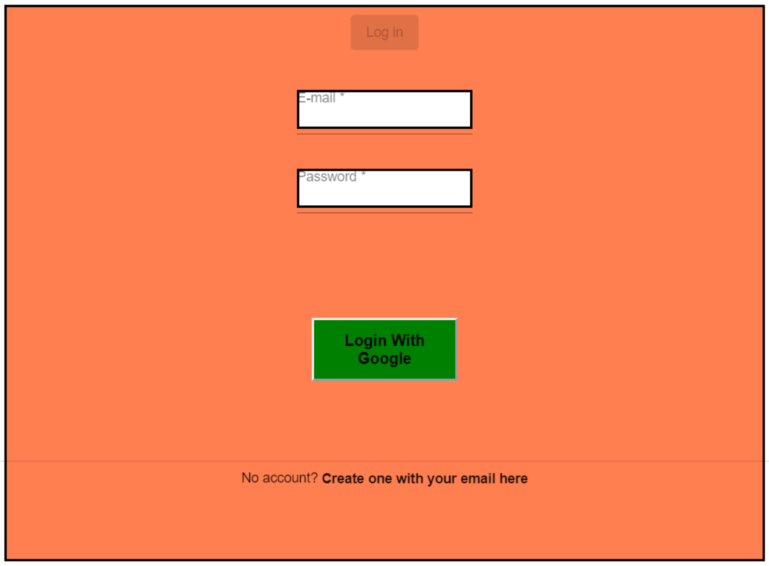
The web-application was designed with the intentionally of being easy to understand and highly navigable. For this the first section of the web-application to be implemented was the header component. The header component consists of a numbers of features, the name of the web-application is bold and centered as would be typically expected of a web-application of this type. Along with this are two dynamic features the page name and page icon, which change depending on which page the user is accessing. After this the header component features five static icons, An icon to link to the products page, an icon to route to user page (colour of icon is dynamic, changing as user login status changes), an icon that links to the user cart (badge displaying count changes as items entered changed), an icon linking an about page to describe the intention of the application and an icon routing to admin panels.

Figure 4.2: Header component as is found in the web-application



When first accessing the web-application users are routed to the ”login” component. The ”Login” component consists of an Angular Material form with two inputs, E-mail and Password to be used for logging in using traditional methods, a button for logins using a Google account and a link to redirect users who have yet to register to the ”signup” component. The ”signup component” consists of an Angular Material form featuring inputs for an e-mail and password. Both of these components feature a login button that is disabled until the input values in the form are considered to be valid, the restrictions on what constitutes a valid input are outlined when an error in input is made.

Figure 4.3: Login component as is found in the web-application



After completing the login or registering as a new user, a standard user has access to four different components. The about page, the product listings page, the user information page and the user cart page. If an non-logged in user tries to access any of these pages then they are automatically re-routed to the login component. This was doing using a standard auth-guard-service.

Figure 4.4: Signup component as is found in the web-application

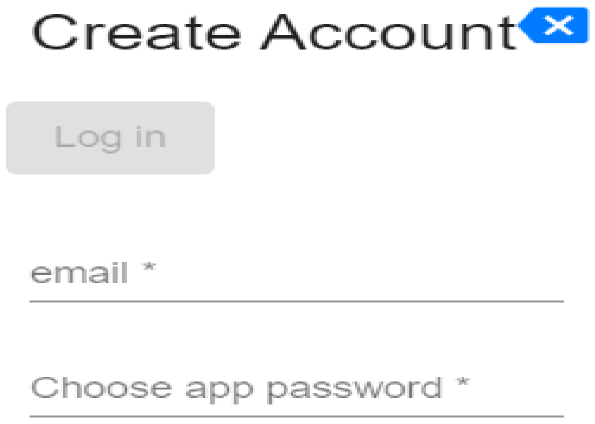
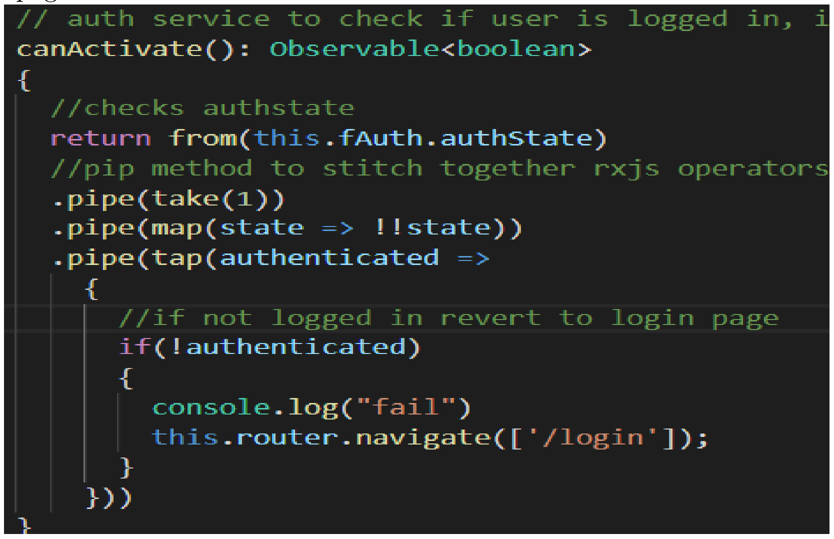


Figure 4.5: Auth-Guard-Service used to stop non logged in users accessing specified pages

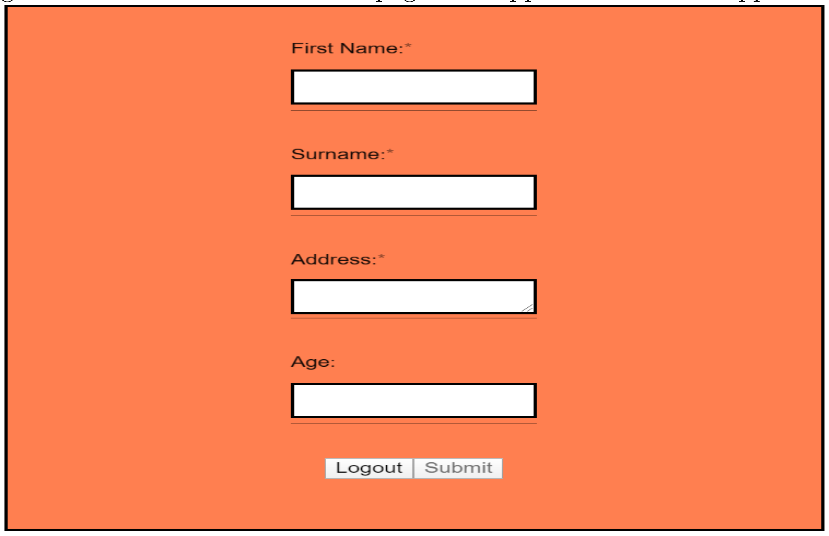


```
// auth service to check if user is logged in, i
canActivate(): Observable<boolean>
{
  //checks authstate
  return from(this.fAuth.authState)
  //pip method to stitch together rxjs operators
  .pipe(take(1))
  .pipe(map(state => !!state))
  .pipe(tap(authenticated =>
    {
      //if not logged in revert to login page
      if(!authenticated)
      {
        console.log("fail")
        this.router.navigate(['/login']);
      }
    }))
}
```

The about component is a simple HTML page, it features no back end functionality and is in place only to inform those that are not aware of the purpose of the web-application or for those interested in seeing which technologies were used in the creation of the web-application. The page consists of a small paragraph about the application and its developer and a series of links to the technologies

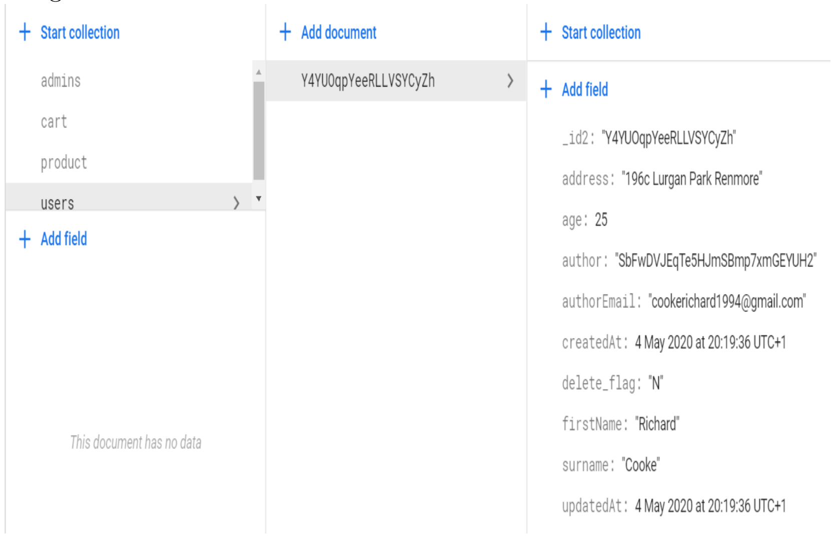
used and the the github repository of the project.
The user information page contains an angular material form that has four input fields (First name, surname, address and age). The first three inputs are given string values and the age field is given a number value. When these fields are correctly filled, the submit button which was initially disabled becomes ready for use. When the submit button is followed the information in the fields of the form, along with the email of the current user and the unique id assigned to the user is sent to the Firebase database as part of a collection of users. Also present on the user information page is a logout button that allows for the current user to be logged out if it is selected, the user is then routed back to the login page.

Figure 4.6: The User Information page as it appears in the web-application



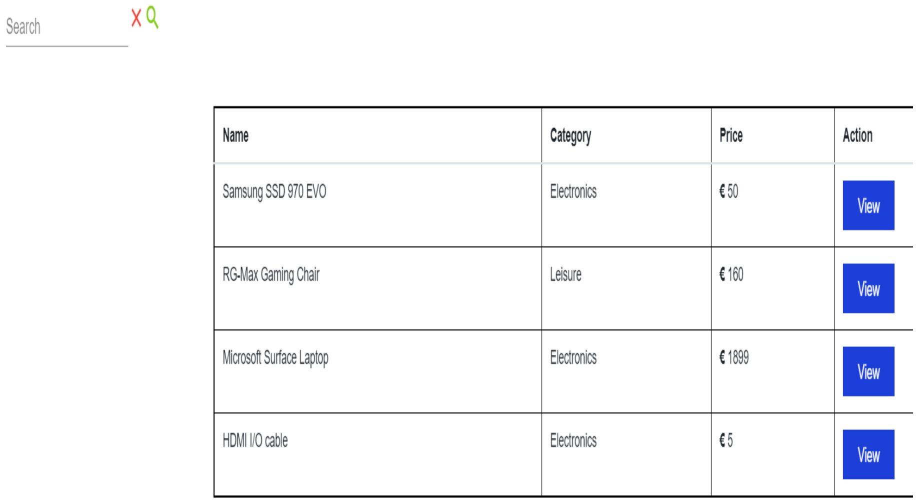
The product listing page is made using two ng-templates. When the user first enters this page they are presented with a search bar on the top left corner(The functionality of the search bar however does not work correctly) and a listing of all the products stored within the Firebase database. Each row of the table constitutes a different product of the database. This was done by using a for loop to asynchronously loop through the data base create a row for for each document. The tables features four columns, the name of the product, the category of the product, the price of the product and a ”view” button that changes to the alternative ng template of the page.

Figure 4.7: User collection and user document in Firebase database



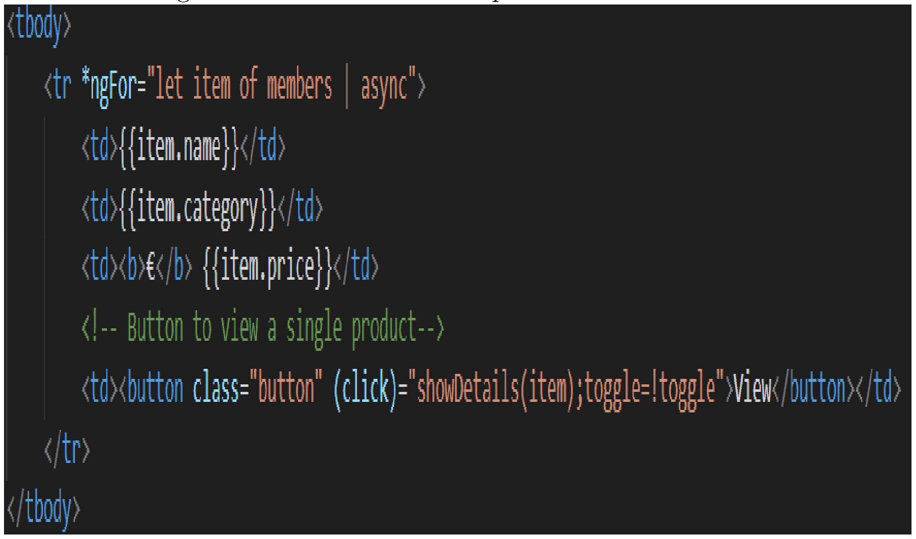
```
_id2: "Y4YUOqpYeeRLLVSYCyZh"
address: "196c Lurgan Park Renmore"
age: 25
author: "SbFwDVJEqTe5HJmSBmp7xmGEYUH2"
authorEmail: "cookerichard1994@gmail.com"
createdAt: 4 May 2020 at 20:19:36 UTC+1
delete_flag: "N"
firstName: "Richard"
surname: "Cooke"
updatedAt: 4 May 2020 at 20:19:36 UTC+1
```

Figure 4.8: The table of products as it appears to the user



Name	Category	Price	Action
Samsung SSD 970 EVO	Electronics	€ 50	View
RG-Max Gaming Chair	Leisure	€ 160	View
Microsoft Surface Laptop	Electronics	€ 1899	View
HDMI I/O cable	Electronics	€ 5	View

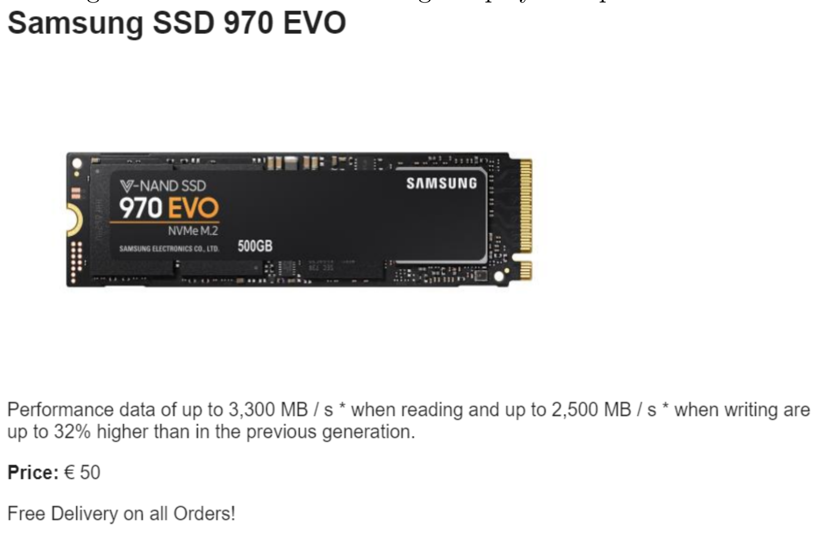
Figure 4.9: How the table of products is constructed



```
<tbody>
    <tr *ngFor="let item of members | async">
        <td>{{item.name}}</td>
        <td>{{item.category}}</td>
        <td><b>€</b> {{item.price}}</td>
        <!-- Button to view a single product-->
        <td><button class="button" (click)="showDetails(item);toggle=!toggle">View</button></td>
    </tr>
</tbody>
```

The alternative ng-template which is called by the user selecting the aforementioned view button displays a single item display for the product that the user has selected the view button for. On this template the user is presented with all of the fields available in the previous templates table along with new information such as a long form description of the product an image of the product. Along with this is an ”Add to Cart” button that uses back end functionality to pass the data of the product in view to the user cart and iterates the badge counter on the cart icon present on the web-applications header. Along with this is a review system in place for each product. Reviews are based on a ”5 star” format where each user can rat the product from 0.5 to 5.0 stars. After the user has rated the product a review becomes present on screen featuring the unique ID of the user that posted the review and the rating to which they applied to the product. Above this the average of all the user reviews is displayed. If no review has been completed for a product then users will be met with a message informing them of such.

Figure 4.10: Product with image displayed in product view



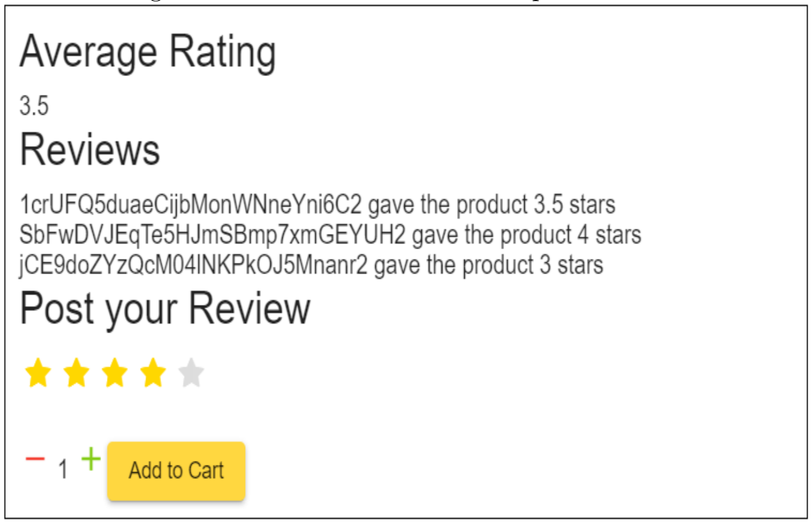
Samsung SSD 970 EVO

Performance data of up to 3,300 MB / s * when reading and up to 2,500 MB / s * when writing are up to 32% higher than in the previous generation.

Price: € 50

Free Delivery on all Orders!

Figure 4.11: Review section of the product view



Average Rating

3.5

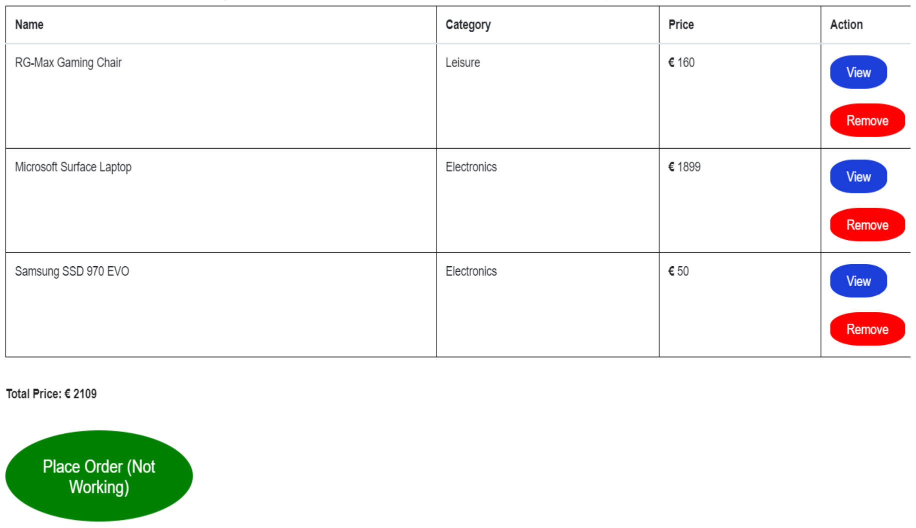
Reviews

1crUFQ5duaeCijbMonWNneYni6C2 gave the product 3.5 stars
SbFwDVJEqTe5HJmSBmp7xmGEYUH2 gave the product 4 stars
jCE9doZYzQcM04INKPkOJ5Mnanr2 gave the product 3 stars

Post your Review

The cart page provides a similar view as the products page. All products within the cart are organized using a table. All the fields that were present in the products table are also present in the cart table along with the addition of a "Remove" button. The remove button allows for the selected product to be removed from the users cart. When an item is removed from the cart, the badge counter on the cart icon of the header is decremented by one. While all products when added to the cart are added to the same collection in the database, the user cart only displays items added by the current user, identified by the Firebase provided unique ID. At the bottom of the cart is the total price of all the items in the cart, along with this is a place order button which is not functional.

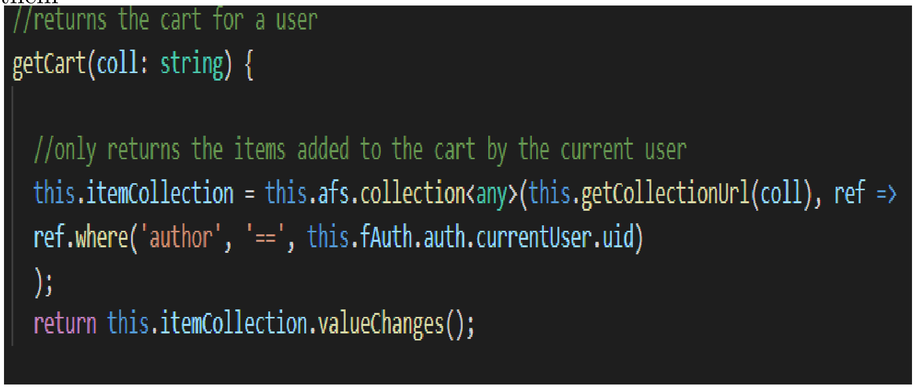
Figure 4.12: How the cart is seen by the user



Name	Category	Price	Action
RG-Max Gaming Chair	Leisure	€ 160	View Remove
Microsoft Surface Laptop	Electronics	€ 1899	View Remove
Samsung SSD 970 EVO	Electronics	€ 50	View Remove

Total Price: € 2109

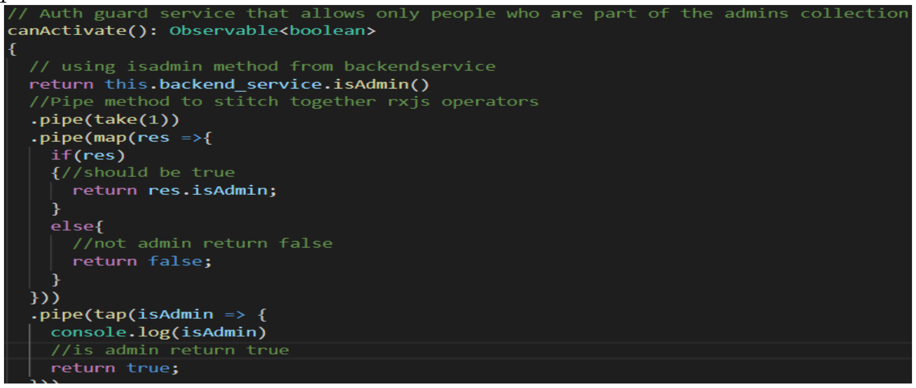
Figure 4.13: How the items of the cart are called where the current user added them



```
//returns the cart for a user
getCart(coll: string) {

  //only returns the items added to the cart by the current user
  this.itemCollection = this.afs.collection<any>(this.getCollectionUrl(coll), ref =>
  ref.where('author', '==', this.fAuth.auth.currentUser.uid)
  );
  return this.itemCollection.valueChanges();
```

The cart page also shares the view option that is present in the products page. This view however is more limited than the view of the products page for an individual item. There is no option to review the item or add it to cart and instead this view is more for the user to ascertain the correct product is in the cart.
The final component available on the web-application is the admin panel. The admin panel is the subject of heavily restricted access, only users who have been granted access to the admins collection on the Firebase database can view this page. The assignment of admin status is done manually by those with access to the database. This restriction of access is accomplished by implementing a similar service to the auth-guard-service to restrict non-logged in users accessing pages. This time an admin-auth-guard-service is used, which provides tighter security.

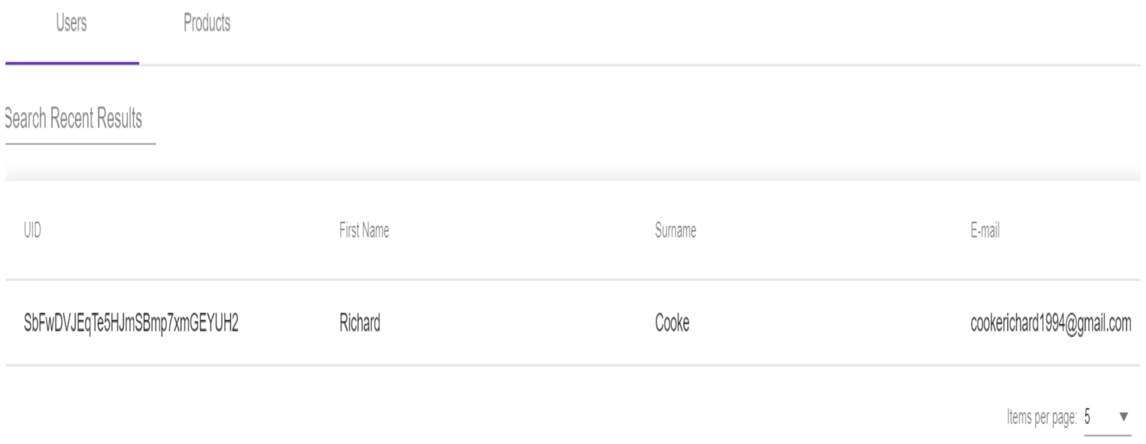
Figure 4.14: Admin Guard Service to prevent non Admins accessing Admin panel



```
// Auth guard service that allows only people who are part of the admins collection
canActivate(): Observable<boolean>
{
  // using isadmin method from backendservice
  return this.backend_service.isAdmin()
  //Pipe method to stitch together rxjs operators
  .pipe(take(1))
  .pipe(map(res =>{
    if(res)
    {//should be true
      return res.isAdmin;
    }
    else{
      //not admin return false
      return false;
    }
  }))
  .pipe(tap(isAdmin => {
    console.log(isAdmin)
    //is admin return true
    return true;
```

The Admin page consists of two tabs, the users tab and the products tab. The users tab returns a full list of users who have entered personal information using the user information page. This data is displayed in table format and features columns relating the the users unique ID, the users First Name, the users Surname and the users E-mail. The Admin has no control over this user data other than viewing privileges. The admin is able to sort the users list by the unique ID and is also able to change the amount of users on display using the paginator.

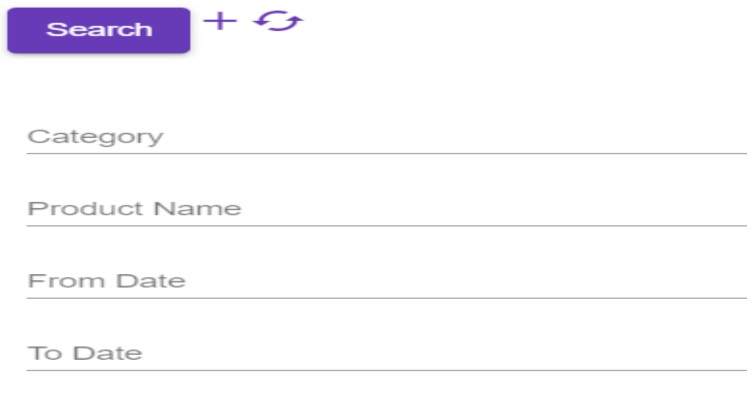
Figure 4.15: Admin view of users list



UID	First Name	Surname	E-mail
SbFwDVJEqTe5HJmSBmp7xmGEYUH2	Richard	Cooke	cookerichard1994@gmail.com

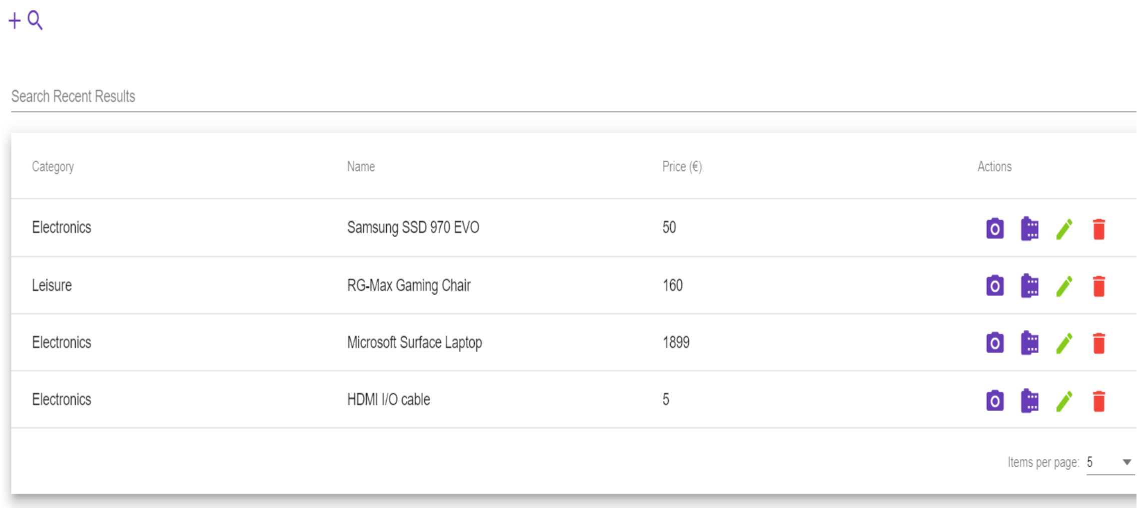
The second tab, the products tab allows for most of the administrator functionality. Here the Admin can views a list of all products in the database in table format, can add new products, can edit products, can delete products, filter through the products and add or remove a picture to the product. There are four templates to the admin products view. Initially the view is set to the search mode were the admin can search through the products with a number of methods, included by narrowing the date using a ate picker although this is not fully functional.

Figure 4.16: Admin view of searching products



The next view is the list of products which is a table format and uses icons located at the end of the product for altering or removing the product in question.

Figure 4.17: Admin view of products list



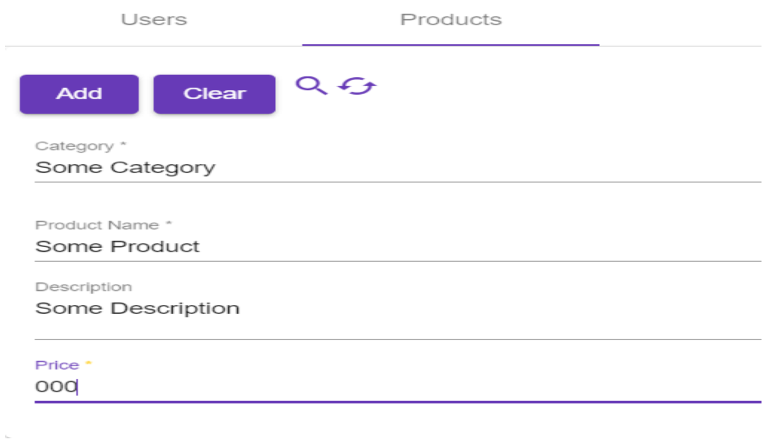
Search Recent Results

Category	Name	Price (€)	Actions
Electronics	Samsung SSD 970 EVO	50	
Leisure	RG-Max Gaming Chair	160	
Electronics	Microsoft Surface Laptop	1899	
Electronics	HDMI I/O cable	5	

Items per page: 5

The Edit mode is accessed by selecting the pencil icon at the end of each product in the list view, it provides 4 fields of entry from an Angular material form where the previous data is carried across, an add button is used to update the information and a clear removes all information present in the data fields. The add mode uses the same view as the edit mode, except that the fields are initialized as empty and the add button creates a new product instead of updating an existing one.

Figure 4.18: Admin view of adding or editing a product



Category *
Some Category

Product Name *
Some Product

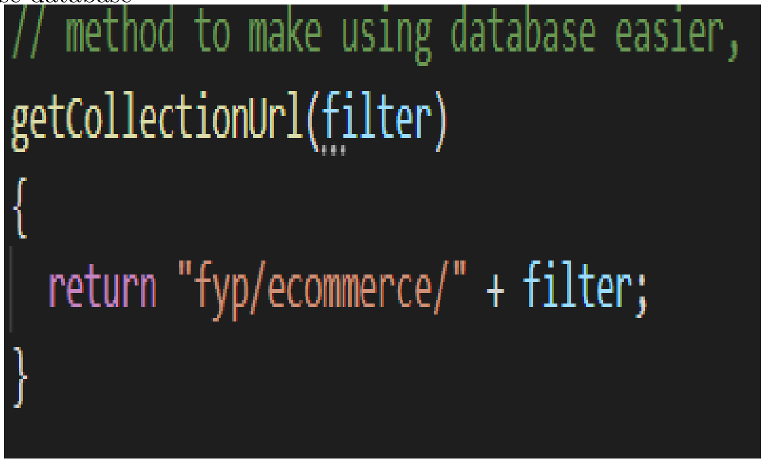
Description
Some Description

Price *
000

4.1.2 Functionality

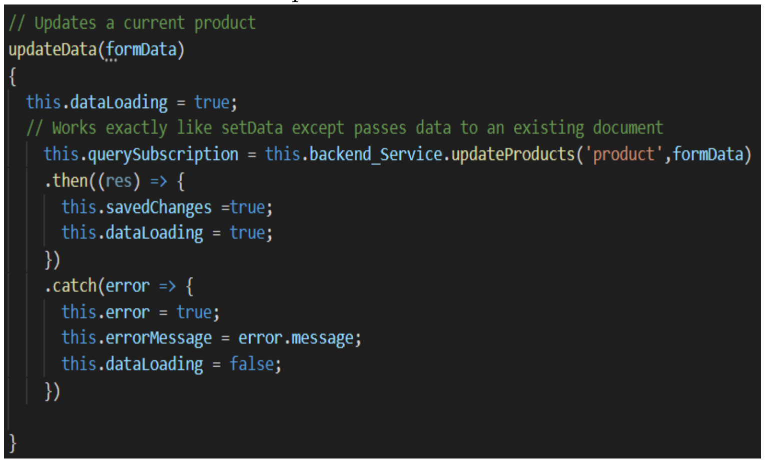
In this section we will look at the services within the application that act as a back end to the web-application. As previously mentioned, each component in Angular has a HTML file which provides the view and a typescript file which carries the pages functionality. I the case of this project most of the actual functionality is present in a service called backendservice.ts. This file is what links the angular application to the Firebase Database using the environment of the project. Backendservice.ts is made up mostly of reusable functions that are called by the typescript files of the Angular components. This done by importing the backendservice as part of the constructor for these typescript files.
Most queries of the database are made by using a hard coded string that dives into a certain level of the Firebase database before entering the a collection that has been specified through a method that has been passed to it. The name of the collection that is to be queried is passed by the typescript file of the angular component on to the back end service which then appends it to the constant string and then sends a request to the database using this newly formed string.

Figure 4.19: getCollectionUrl method is used in every method that queries the Firebase database

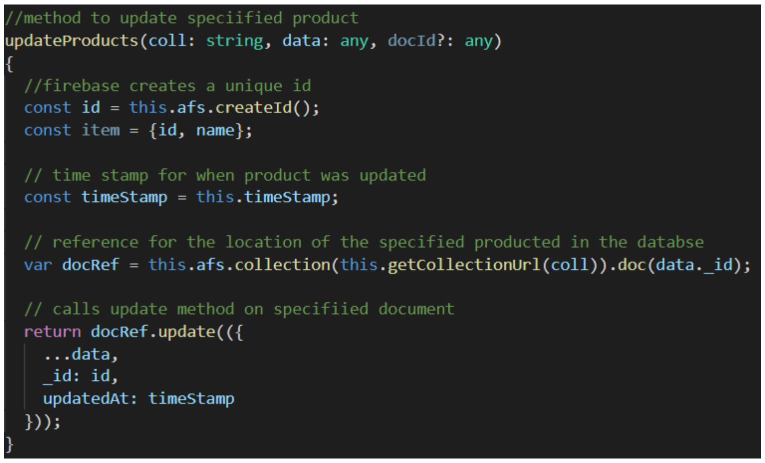


```
// method to make using database easier,
getCollectionUrl(filter)
{
  return "fyp/ecommerce/" + filter;
}
```

Figure 4.20: For example the updateData method of setProducts.ts makes use of the backendservice method updateProducts



```
// Updates a current product
updateData(formData)
{
  this.dataLoading = true;
  // Works exactly like setData except passes data to an existing document
    this.querySubscription = this.backend_Service.updateProducts('product',formData)
    .then((res) => {
      this.savedChanges =true;
      this.dataLoading = true;
    })
    .catch(error => {
      this.error = true;
      this.errorMessage = error.message;
      this.dataLoading = false;
    })

}
```

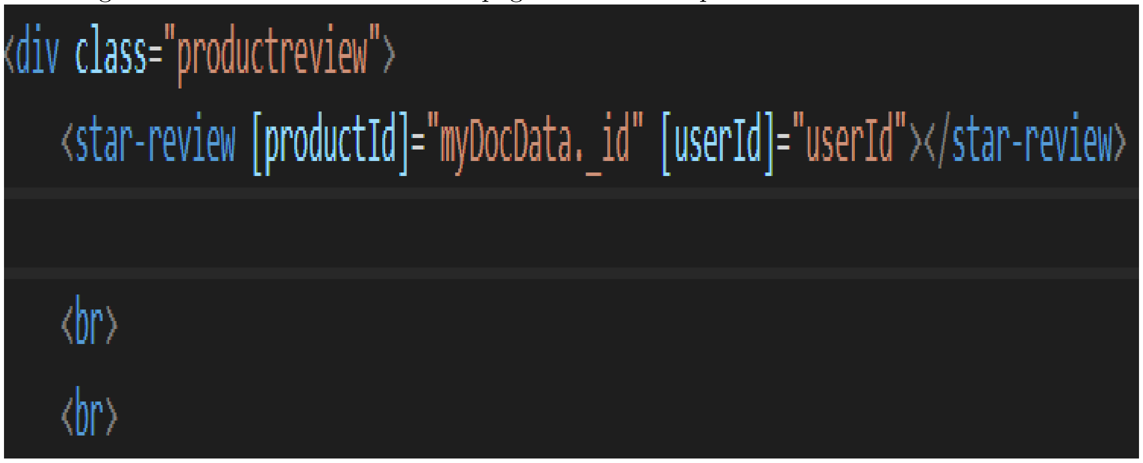


```
//method to update speciified product
updateProducts(coll: string, data: any, docId?: any)
{
  //firebase creates a unique id
  const id = this.afs.createId();
  const item = {id, name};

  // time stamp for when product was updated
  const timeStamp = this.timeStamp;

  // reference for the location of the specified producted in the databse
  var docRef = this.afs.collection(this.getCollectionUrl(coll)).doc(data._id);

  // calls update method on speciified document
  return docRef.update(({
    ...data,
    _id: id,
    updatedAt: timeStamp
  }));
}
```

Review Functionality

The functionality of using the review system is similar to creating a relationship like in an SQL databases with a many to many relationship. Unlike the other collections, it was determined to make the star collection a root collection which could dip into both the user collection and the product collection. This was done to establish the relationship of the collections. The review functionality is done by using an angular component as an inner-page and passing it two variables, the ID of the product being reviewed and the user ID of the reviewer, both of which are present when on the single product view.

Figure 4.21: How the inner review page is called into product.html



```
<div class="productreview">
   <star-review [productId]="myDocData._id" [userId]="userId"></star-review>


   <br>
   <br>
```

Upload Image Functionality

Like the Star Review page, the upload page is called as an inner page, this time as part of the set products page to be handled by the administrator. The upload page takes two parameters, the path of the file or image being uploaded and the id of the document this path is to be added to. Most of the functionality considered with file upload is adapted from the angular 2 handbook, which is referenced within the filesize.pipe.ts file which is used to handle the mathematics involved in file upload. The image upload occurs within the set product as one of the actions the administrator can take when creating or editing a product. When a product does not have an attached image the photo-reel icon is blacked out. After an image has been attached the blacked out image-reel is replaced with two icons a camera icon for viewing the uploaded image and a differently coloured photo-reel icon that when followed, removes the attached image of the specified document.

Figure 4.22: Top: After an image is uploaded, Bottom: Before an image is uploaded.

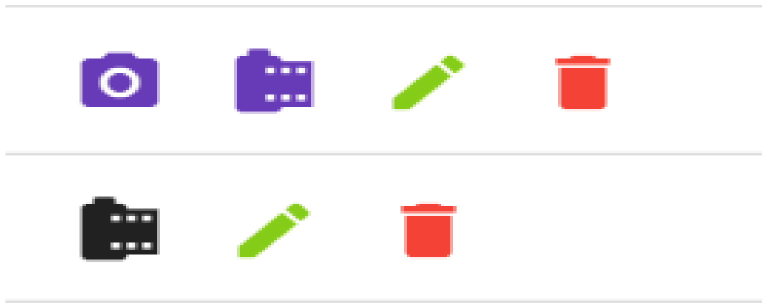
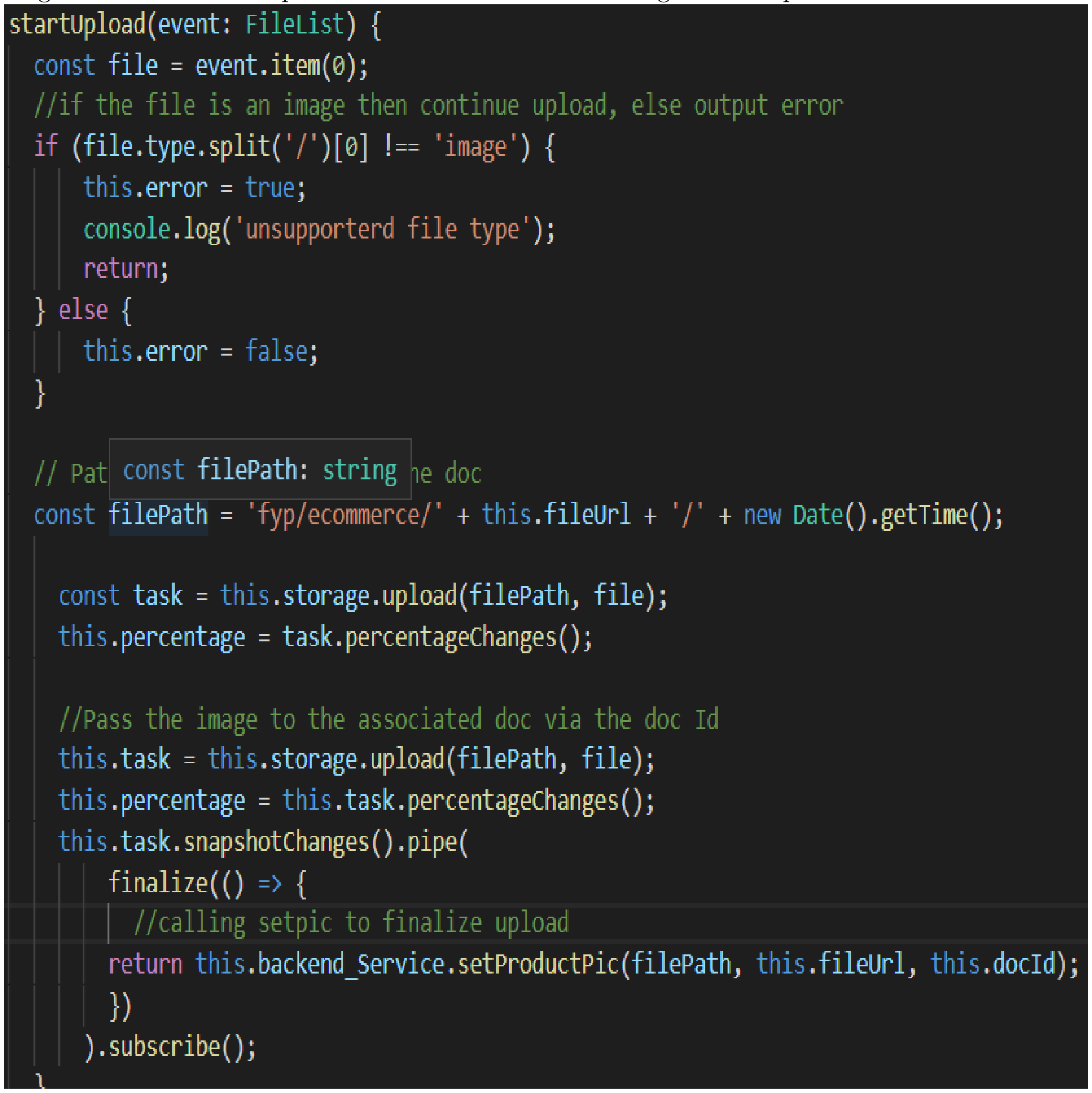


Figure 4.23: The startUpload method used to add an image file to a product



```
startUpload(event: FileList) {
  const file = event.item(0);
  //if the file is an image then continue upload, else output error
  if (file.type.split('/')[0] !== 'image') {
      this.error = true;
      console.log('unsupporterd file type');
      return;
  } else {
      this.error = false;
  }

  // Pat const filePath: string ne doc
  const filePath = 'fyp/ecommerce/' + this.fileUrl + '/' + new Date().getTime();

    const task = this.storage.upload(filePath, file);
    this.percentage = task.percentageChanges();

    //Pass the image to the associated doc via the doc Id
    this.task = this.storage.upload(filePath, file);
    this.percentage = this.task.percentageChanges();
    this.task.snapshotChanges().pipe(
        finalize(() => {
          //calling setpic to finalize upload
        return this.backend_Service.setProductPic(filePath, this.fileUrl, this.docId);
        })
      ).subscribe();
```

Chapter 5

System Evaluation

This chapter will outline what methods for testing the functionality of the web-application took place. It will also examine whether the earlier outlined requirements of the application were met or not and will discuss the limitations of the technologies used.

5.1 Testing

Testing of the web-application was performed in a number of ways. Throughout the development cycle of the applied project Test Driven Development was implemented, this means that before a new unit of code was started the preceding unit was developed into a fully functioning state via testing, this was done so as to stop a build up of bugs and when being paired with the Extreme Programming methodology adapted early on allowed for most of the core functionalities outlined to be integrated without impacting the application in a negative way.

5.1.1 Usability Testing

In this case the testing was carried out in 3 stages. In the first stage, fellow participants in the course were able to use completely local version of the project that had limited functionality. This stage of testing was to ascertain that the web-application had the desired feel of an expected e-commerce application.
In the next stage of usability testing, students in the module were given a version of the project that had a working connection to the Firebase database along with most basic functionality associated with the web-application. This stage of testing was done for more feedback of the applications feel and to catch any bugs. It was at this stage that a student reported a bug in the search products function.
The final stage of usability testing was to have these same students access the deployed web-application and to provide general feedback. These tests were carried out by students only.

5.1.2 Integration Testing

In the case of this project integration was carried out at the completion of each unit of work. When the functionality had been developed to a level deemed working it was integrated into the system to see what impact it had on the system as a whole or on any previously integrated components of the application. These tests were carried out by the developer only.

5.1.3 Compatibility Testing

In the case of this project compatibility testing was conducted by deploying the application onto multiple different browsers and to different Operating System environments. These tests were conducted by both the developer and other students. The browsers using in this testing phase were:

- Google Chrome (The main test base)
- Mozilla Firefox
- Microsoft Edge

The Operating System environments that were used in this testing phase were:

- Windows 10
- Windows 8
- Manjaro

The application deployment was also tested on mobile browsers and deployed successfully, however because it had not been configured to adapt to the screens of these devices the placement of the UI was not ideal.

5.1.4 Acceptance Testing

This testing was conducted at the same time as the third stage of Usability testing and was to ascertain whether the application met the standards expected of an e-commerce application and was determined by the feedback provided by the student involved.

5.2 Limitations

One of the key limitations uncovered in within this project was that of the compatibility of Firebase with some deployment sites. The original plan when intending to develop this application was to use Firebase for authentication and for database storage and to use heroku as a means to host the application on a cloub platform. However, this changed late into the development cycle when attempting to deploy an angular firebase based application to heroku numerous problems where encountered and with a lack of documentation relating to pushing an application of this type without a back end server as is not used in firebase databases the original plan was abandoned. Luckily, this situation was resolved as firebase had its own deployment and hosting service which was free. However, this lack of cooperation between a firebase database and other platforms is something that should be classified as a limitation of the technology. On the limitations of the applications, it was never tested to a breaking point with automated testing where 100's of requests would be sent through the application ever few seconds, it did receive a minor stress test were the author and 7 fellow student spammed requests towards the applications database and it experienced no falter as would be expected.

Chapter 6

Conclusion

In this chapter, the author will discuss the conclusions drawn throughout the projects development and how or if the initial goals laid out in the introduction of this paper were met.

6.1 Overview

In the introduction of this paper the objectives of the projects were specified as:

- Produce a simple, easy to use and understand, web application
- Gain further understanding of web technologies
- Create an e-commerce web application that meets modern expectations
- Complete the given assignment as a lone venture
- Introduce the concept of the project
- Provide an understanding of web technologies
- Outline the development of the applied project

It can be determined that the main goal of this project had been for the web-application that was produced to have had a simple nature and to be easy to use and understand. Given the feedback the author received from the acceptance testing section of the system evaluation chapter it is of the authors opinion to conclude this goal as a success.

The author concludes that the goal of gaining a further understanding of web technologies has also been achieved as the author has developed a functioning application which utilizes technologies such as Firebase which had not been used by the author previously and the work completed within the applied project is proof of furthering their knowledge in using technologies such as Angular and Node with which they had previous experience with.

The goal of making an e-commerce web-applications was met for the most part, although feedback provided from fellow students was positive for the most part it was almost consensus among those that partook in the Usability and Acceptance testing stages that an e-commerce application such as this should be able to provide the functionality to process a users order.

In the opinion of the author, completing the assignment as a solo venture was also a success. Although the author was originally to undertake this project as a team exercise, the realization that the scope of the project was not grand enough to be completed by a 3-person party led to the original team to be dissolved. However, the research conducted during this time was still applied to this project as the scope of the project and its them stayed largely intact. Completing the project and meeting the user requirements outlined in the introduction to this paper, of what was originally intended to be approached by a group rather than a single person, lead to the conclusion that this goal was a success.

The author feels that throughout the introduction of the project that the concept of the project was correctly introduced to potential readers and is presented in a way that is understandable to those without heavy experience in the area explored.

The author can conclude that readers where suitably provided with an understanding of the web technologies implemented in the development of this project. This is concluded from the third chapter of this paper wherein the author extensively explains which web technologies are used and how they are used with the application.

The author feels that the development of the web-application was outlined in a fine manner as the chapter on the System design highlights the construction of all the components within the web application and discuss the back end technologies employed too.

6.2 Future Modifications

It is the opinion of the author that the application has numerous routes in which it can grow. Firebase authentication allows for user authentication to be provided from many sources, at the moment the application allows for application to be accessed for users with Google accounts and for classic email and password inputs, however Firebase allows for authentication through almost any social platform, by phone or even by Github account.

Another future modification is the use of higher tiers of data storage from Firebases' servers. Currently the application only employs the use of the free tier of Firebases' database service, for a fee this can be upgraded to faster, more responsive servers.

Aside from this, some obvious modification can be made to rectify the lack of order placement functionality and the faulty search functionality within the application. It can also be seen that a more thorough, professional styling can be applied throughout the application and that it can be geared to words looking presentable on mobile devices as was outlined in the feedback of some of the testing.

6.3 Final Conclusion

Overall the author feels like the experience of the development of the application was a strong learning experience. The weekly meetings with the project supervisor provided a pseudo-customer experience to the assignment not present in other modules on the course. The author also feels that the module allowed for a significant amount of self learning, which is something the course has always pushed as something the participants should strive for.

6.4 Final Words

As the author prepares to submit this document and bring a close to the majority of work in their final year of education at GMIT, it is a pleasant experience to be able to compare the application delivered as part of the project compared to the similar project they were tasked with in their first year of education at GMIT and be able to see solid proof of their growth as a software developer.

References

[1] *A Comparative Analysis of NodeJs.* URL: https://repository.stcloudstate.edu/cgi/viewcontent.cgi?article=1004&context=csit_etds.

[2] *Advantages and Disadvantages of Choosing Firebase.* URL: https://www.altexsoft.com/blog/firebase-review-pros-cons-alternatives/.

[3] *Advantages and Disadvantages of NodeJs.* URL: https://codewithhugo.com/node-pros-and-cons/.

[4] *Angular Version 9.* URL: https://blog.angular.io/version-9-of-angular-now-available-project-ivy-has-arrived-23c97b63cfa3.

[5] *Angular: Advantages vs Disadvantages.* URL: https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/.

[6] *AngularCLI Documentation.* URL: https://angular.io/cli.

[7] *Extreme Programming.* URL: https://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1).

[8] *Firebase Authentication.* URL: https://firebase.google.com/docs/auth.

[9] *Firebase Deployment.* URL: https://firebase.google.com/docs/hosting/deploying.

[10] *Firebase Vs Traditional Infrastructure.* URL: https://www.academia.edu/37015157/Traditional_Infrastructure_vs._Firebase_Infrastructure.

[11] *Javascript Documentation.* URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[12] *Node Package Modules.* URL: https://docs.npmjs.com/about-npm/.

[13] Alen Šimec and Magličić. "Comparison of JSON and XML Data Formats". In: Sept. 2014.

[14] *Single-Page Vs Multi-Page.* URL: https://hackernoon.com/single-page-applications-the-rise-of-web-apps-in-2020-un6c32gm.

[15] *Typescript Documentation.* URL: https://www.typescriptlang.org/docs/home.html.