

**A DYNAMIC WEB APPLICATION FOCUSED ON NICHE PRODUCT
BY**

Md. Shamim Reza Mission

ID: 182-15-11638

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Abbas Ali Khan
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Mohammad Kaosain Akbar
Lecturer
Department of CSE
Daffodil International University



**DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
JANUARY 2022**

APPROVAL

This project titled “**A Dynamic Web Application Focused on Niche Product**”, submitted by **Md. Shamim Reza Mission**, ID No: **182-15-11638** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **06-01-2022**.

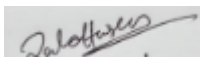
BOARD OF EXAMINERS

Chairman



Dr. Touhid Bhuiyan
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Internal Examiner

Zahid Hasan (ZH)
Associate Professor

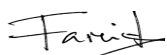
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Internal Examiner

Mohammad Monirul Islam (MMI)
Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Dr. Dewan Md. Farid

Professor

Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

I hereby declare that this internship has been done by me under the supervision of **Md. Abbas Ali Khan, Senior Lecturer, Department of CSE**, Daffodil International University. I also declare that neither this internship nor any part of this internship has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

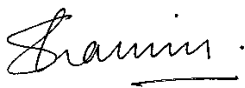


Md. Abbas Ali Khan
Senior Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Mohammad Kaosain Akbar
Lecturer
Department of CSE
Daffodil International University

Submitted by:



Md. Shamim Reza Mission
ID: 182-15-11638
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to Almighty God for His divine blessing making it possible for me to complete the final year internship successfully.

I am grateful and wish my profound indebtedness to **Md. Abbas Ali Khan, Senior Lecturer**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of Computer Networking and Data Communication to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to thank other faculty members and the staff of the CSE department of Daffodil International University.

I would like to thank our entire coursemate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

In today's fast-changing business environment, where thousands of products e-commerce stores are competitive there comes the solution "Niche Product Web Application". The main focus of this project is to create a suitable e-commerce web application for a single type of product-based business. It's a way to target those individual people who are mainly passionate or like to find a single type of product in a place. Day by day people is adopting this type of service. In this project, I mainly have two modules. One is for customer/user and another is for admin. Normally users can view all products with or without registration and must have to log in if they want to buy/book/order something from the website. Another most important part of this project is Dashboard. From there an admin can add or remove products anytime they want, and also can cancel or confirm any order made by users. Also, the admin can add any existing user or new use as an admin. Another option is available for users that they can easily manage their orders and can add any review which will show on the home page. Users can confirm their order by completing their payment using any International Credit Cards.

This is a MERN (JavaScript software stack for building dynamic websites and web applications) Stack-based project which stands for MongoDB, Express.js, React JS, and Node.js. I used React Js (Front-end JavaScript library), Material UI (React UI library), Firebase (Authentication, Authorization, and Hosting), MongoDB (Cross-platform document-oriented No SQL Database), Node.js (Back-end JavaScript runtime), and Express.js (Back-end web application framework for Node.js), Stripe (Online payment processing for internet businesses) To develop this project.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	iii
Acknowledgments	iv
Abstract	v
CHAPTER	
CHAPTER 1: Introduction	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Expected Outcome	2
1.5 Methodology Development Model	3
CHAPTER 2: Background	4-9
2.1 Related Works	4
2.2 System Requirement	4
2.2.1 System Requirement Specification	4
2.2.2 Hardware Interfaces	4
2.2.3 Software Interfaces	5
2.3 Technologies	5

2.3.1 HTML	5
2.3.2 CSS	6
2.3.3 JavaScript	6
2.3.4 React	7
2.3.5 Material UI (MUI)	7
2.3.6 Node.js	8
2.3.7 Express.js	8
2.3.8 Firebase	8
2.3.9 MongoDB	9
2.3.10 Heroku	9
CHAPTER 3: Implementation	10-12
3.1 Use Case Diagram	10
3.2 Use Case Description for Web Application	11
CHAPTER 4: Design Specification	13-23
4.1 Front End Design	13
4.1.1 Home Page	13
4.1.2 Explore Page	15
4.1.3 Registration Page	16
4.1.4 Login Page	16
4.1.5 Order Modal	17
4.1.6 User Dashboard	17

4.1.7 Admin Dashboard	19
4.2 Back-End Design	21
4.2.1 MongoDB	21
4.2.2 Firebase	22
4.2.3 Heroku	23
Chapter 5: Conclusion and Future Scope	24
5.1 Discussion and Conclusion	24
5.2 Scope for Further Career	24
References	25

LIST OF FIGURES

FIGURES	PAGE
Figure 1.5.1: Methodology Development	3
Figure 3.1.0: Use Case Diagram	10
Figure 4.1.1.1: Home Page-1	13
Figure 4.1.1.2: Home Page-2	14
Figure 4.1.2.1: Explore Page	15
Figure 4.1.2.2: Log in Modal	
Figure 4.1.3.1: Register Page	16
Figure 4.1.4.1: Login Page	16
Figure 4.1.5.1: Order Modal	17
Figure 4.1.6.1: User Dashboard	17
Figure 4.1.6.2: Add Review Form	18
Figure 4.1.6.3: Payment Form	18
Figure 4.1.7.1: Manage All Orders Page	19
Figure 4.1.7.2: Manage Products Page	19
Figure 4.1.7.3: Add a Product Page	20
Figure 4.1.7.4: Add an Admin Page	20
Figure 4.2.1.1: MongoDB Login	21
Figure 4.2.1.2: Databases	21
Figure 4.2.1.3: Data Lists	22
Figure 4.2.2.1: Firebase Console	22
Figure 4.2.2.2: Firebase Authentication (User's List)	23
Figure 4.2.3.1: Heroku Dashboard	23

LIST OF TABLES

TABLES	PAGE
Table 3.2.1: User Login and Registration Form	11
Table 3.2.2: Admin Login Form	12

CHAPTER 1

Introduction

1.1 Introduction

A niche website is a website that serves a small part of a large market with a focus on similar, direct interests. Some people want to buy a single product online but get distracted by other products or something, they don't get the actual product of their choice. There are many e-commerce websites on the internet but I tried something different. Which mainly focuses on selected entrepreneurs who want to bring their niche product business on the internet and those users/customers who find a single type of product on the internet. That's why I made this wristwatch web application.

1.2 Motivation

The main motivation of this project is to serve those small business persons or entrepreneurs who want to bring up their business online and those users who get distracted by other products when they want to get a niche product. On the other hand, the online shop provides us with a comfortable life that never provides typical business. For example, we can order any niche product by using our smart devices through the internet. Any time we can visit any shop that we like to visit through an e-commerce website virtually. So, by using the e-commerce website, we can save both saving money and also, we can visit many shops without going shopping physically. If we can use to about e-commerce website many advantages we will get. So, my goal is to provide these services for both of them who are small business people and customers. And after completing this project I'll be able to develop any web application and my web development will surely improve that's why I choose this project.

1.3 Objectives

- **Reduce Management Costs:** An E-commerce website business can easily determine the administrative costs of any business because the virtual system does not require a large number of administrative costs. Proper use of digital marketing reduces marketing costs and becomes tangible marketing costs. Customers can receive their product delivery at home. Significant time will be saved by purchasing through the e-commerce website.
- **Increasing business relationship:** A new company can grow when it sells e-commerce in a short time. Relationships between entrepreneurs and customers will grow soon. The efficient use of modern e-commerce technology is the best way to grow up business in a very short time.
- **Achieving the main goal:** Many entrepreneurs can reach a well-targeted audience in a short period through e-commerce. A large number of customers are now regular Internet users. To find the required product available on the e-commerce website. Every day the community becomes smart and wants the right use of time. Niche product web applications always serve their facilities.

1.4 Expected Outcome

The niche product-based web applications are very popular and beneficial nowadays. It helps new entrepreneurs to bring their business online and target those selected customers who have an interest in shopping online of their selected interest. By completing this project, I want to let them know about this. And also, it will help to develop any kind of dynamic web application and also will enhance my web development knowledge.

1.5 Methodology Development Model

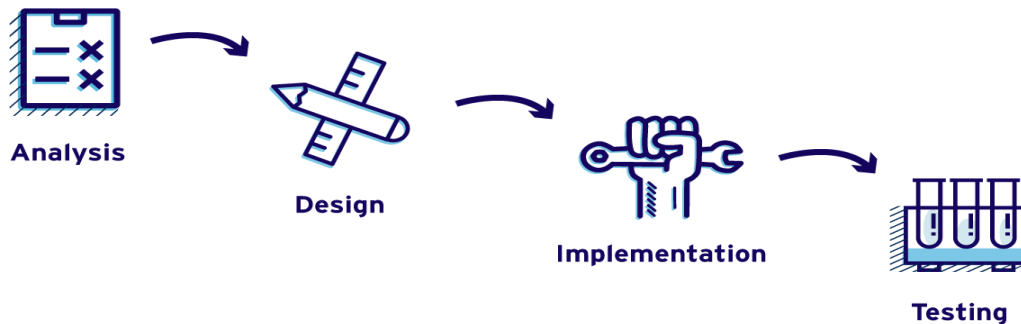


Figure 1.5.1: Methodology Development Model [1]

Requirement Analysis: It is very important to analyze the proper needs and requirements to develop a dynamic web application. It helps to properly design the application as per requirement. And proper functioning of a web application is very important to analyze before designing.

System Design: After analysis, it is time to design the application properly. Because, every web application needs to make a clean, beautiful, and user-friendly system design. Which carries the external beauty of a web application and that makes it unique from other web applications that exist on the internet.

Implementation: After completing the basic objective now it is important to implement everything together. The implementation makes a visual output of each idea.

Testing: To make a better user-friendly interface and error-free web application, testing is very important after implementation. Every available option should be tested from both ends. It helps to find out the developer's mistakes and errors as well as to make an error-free application.

CHAPTER 2

Background

2.1 Related Works

The E-Commerce industry is growing day by day. People can easily purchase anything from online stores. There are many kinds of e-commerce sites available in our country such as Daraz, Evaly, bikroy.com, etc. But these sites offer all kinds of products on their site. But I will make a web application that only serves niche products.

Many people only want that product that they are searching for and they don't want to get distracted by other types of products. So, I think this project might help them and will help newcomer entrepreneurs to bring their business online.

2.2 System requirement

2.2.1 Software Requirement Specification

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs [2].

2.2.2 Hardware Interfaces

- ✓ Processor: Any Latest Model
- ✓ Ram: 2GB or Above
- ✓ Storage: 50GB or Above
- ✓ Input devices: Mouse, Keyboard
- ✓ Output Device: Monitor
- ✓ Internet Connection: Mobile Internet or Broadband

2.2.3 Software Interfaces

- ✓ Editor: Visual Studio Code
- ✓ Browser: Google Chrome
- ✓ Operating System: Windows 10

2.3 Technologies

This is a NoSQL Database based project developed with HTML5, CSS3, React JS, Material UI, Node.js, Express.js, Heroku, Firebase, and MongoDB

2.3.1 HTML

HTML, or Hypertext Markup Language, allows web users to create and edit categories, and links using elements, tags, and attributes. Even so, it is worth noting that HTML is not considered a programming language as it can't create dynamic functionality.

HTML has many usage conditions, namely:

Web development: Developers use HTML code to design how a browser displays features of a web page, such as text, hyperlinks, and media files.

Internet Navigation: Users can comfortably navigate and insert links between related pages and websites as HTML is widely used to embed hyperlinks.

Web documents: HTML makes it possible to edit and format documents, just like Microsoft Word.

It is also worth noting that HTML is now regarded as the official web standard. The World Wide Web Consortium (W3C) maintains and enhances HTML specification, as well as provides regular updates [3].

2.3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of text written in a markup language such as HTML. CSS is the basic technology of the World Wide Web, compatible with HTML and JavaScript.

CSS is designed to enable the segregation of presentation and content, including layout, colors, and fonts. These variations can improve content accessibility, provide more flexibility and control in the segregation of presentation features, enable more web pages to share formatting by specifying the appropriate CSS in a separate .css file that reduces complexity and duplicate content structure and enables .css file to be cached to improve page load speed between file-sharing pages and their formatting.

The separation of formatting and content makes it easy to present the same marker page in different styles in different ways of rendering, such as on-screen, in print, by voice (with a speech-based browser or screen reader), and in Braille-based tactile devices. CSS also has rules for other formats when content is accessed on mobile [4].

2.3.3 JavaScript

JavaScript (/ 'dʒɑ:vəˌskript /), commonly abbreviated as JS, is a programming language compliant with ECMAScript specifications. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, flexible typing, prototype-based object orientation, and first-class functions [5].

Next to HTML and CSS, JavaScript is one of the most basic technologies on the World Wide Web. More than 97% of websites use their client-side in the behavior of a web page, usually incorporating third-party libraries. All major web browsers have a JavaScript engine dedicated to executing the code on the user's device.

Like multi-paradigm language, JavaScript supports event-specific, functional, and essential programming styles. It has application programming interfaces (APIs) for text, dates, common expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as network, repository, or image services. A web browser or other operating system provides JavaScript APIs for I / O.

JavaScript engines were originally used only in web browsers, but have now become an integral part of other software programs, especially servers and various applications.

2.3.4 React

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications [6].

However, react is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

2.3.5 Material UI (MUI)

Material-UI is simply a library that allows to import and use of different components to build user interaction in React applications. This saves a lot of time as developers do not need to write everything from scratch.

Material-UI widgets are strongly influenced by Google's principles in creating user interfaces. Therefore, it is easy for developers to build attractive apps [7].

2.3.6 Node.js

Node.js is an open-source server-side runtime environment built on the Chrome V8 JavaScript engine. Provides an event-based, non-blocking I/O and cross-platform runtime environment to create a very flexible server-side application using JavaScript.

Node.js can be used to build multiple types of applications like command-line applications, web applications, real-time chat applications, REST API servers, etc. However, it is extensively used to build network applications such as web servers, such as PHP, Java, or ASP.NET [8].

2.3.7 Express.js

Express.js, or just Express, is the backbone of the Node.js web application, published as free and open-source software under the MIT license. It is designed to build up web applications and APIs. It is known as the de facto standard server frame for Node.js.

The original author, TJ Holowaychuk, described it as a server inspired by Sinatra, which means it's very small with plenty of features available as plugins. Express is the spine of popular development stacks such as MEAN, MERN, or MEVN stack, as well as the MongoDB database software and the front-end JavaScript framework or library [9].

2.3.8 Firebase

Firebase is a development platform originally known for its real-time architecture which is a multi-node site, value key optimized for data synchronization, generally between user devices or smartphones, and centralized storage in the cloud. It is designed to make life easier for developers with a lot of push-up and data management. That frees app developers from the programming burdens associated with managing versions or locations. They can write new pieces in Firebase and the data will be compatible throughout the system.

Firebase is very important because it can distribute and synchronize changes between local copies of user information on cloud versions with versions stored in the cloud. Firebase

eliminates many mixing challenges to authenticate, sync, and differentiate by integrating multiple versions and ensuring that the appropriate pieces are the same throughout the system [10].

2.3.9 MongoDB

MongoDB is an open-source data management system for NoSQL. NoSQL is used as a substitute for traditional relational databases. The NoSQL database is very helpful in working with large distributed datasets. MongoDB is a tool to manage document-oriented data, storing or extracting data.

MongoDB supports different types of data. It is one of the nonrelational database technologies that emerged in the mid-2000s under the NoSQL proclamation - in general, to be used in big data systems and other processing operations that incorporate data that does not fit the solid relationship model. Instead of using tables and rows as in related websites, MongoDB architecture is made up of collections and texts.

Organizations can use this for their ad-hoc queries, indexing, load balancing, aggregation, server-side JavaScript execution, and other features [11].

2.3.10 Heroku

Heroku is a cloud platform as a service (PaaS) that supports multiple programming languages. One of the first cloud platforms, Heroku has been upgraded since June 2007, when it not only supports Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go [12]. Therefore, Heroku is said to be a multilingual platform because it has the characteristics of developers who will build, run and measure applications in the same way across multiple languages.

CHAPTER 3

Implementation

3.1 Use Case Diagram

Using a case diagram to represent work with pictures. In the context of it, show all the activities for this project. Two actors mainly play the role in this project. One is a user and the other is an admin. Any customer can visit our web application without registration but to purchase any product, the user must require registration and login to the user profile. Many features found on our website will all be displayed in the use case diagram.

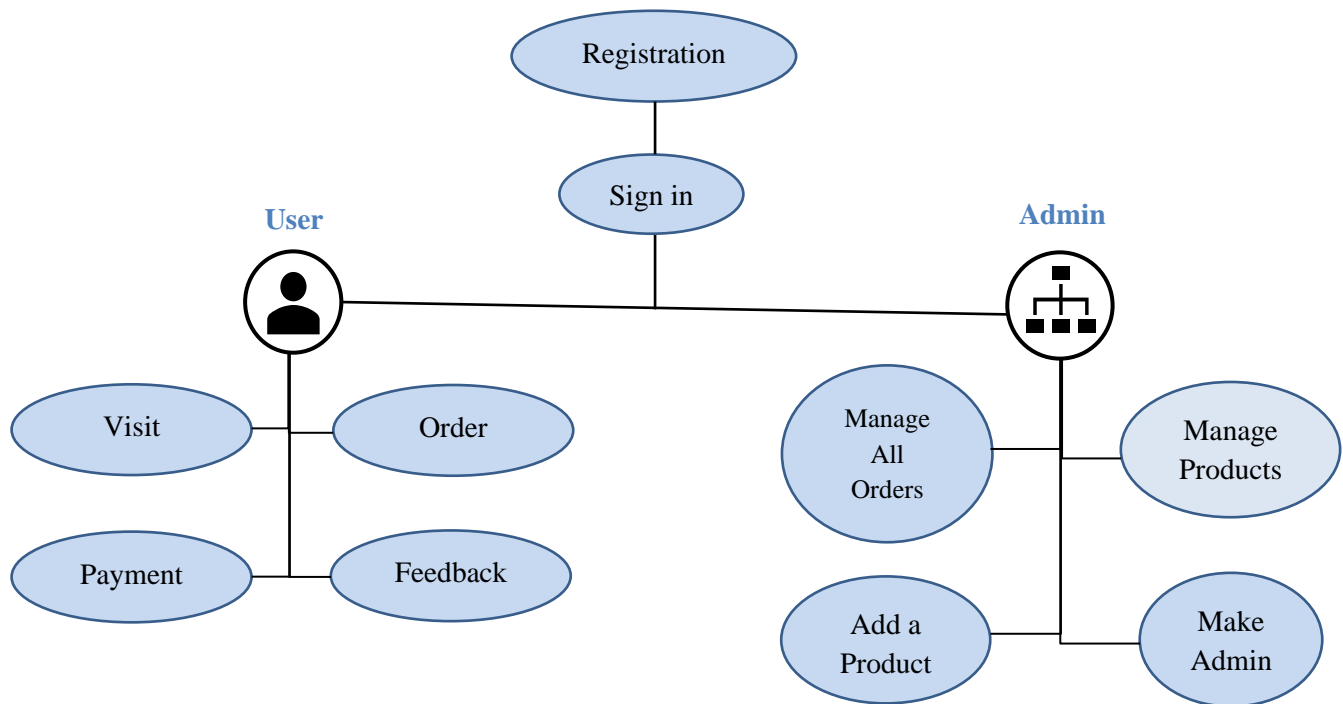


Figure 3.1.0: Use Case Diagram

3.2 Use Case Description of Web Application

Table 3.2.1: User Login and Registration Form

Use case for	User login
Actor	User (registered and non-registered)
General Path	1) Type username/email 2) Type password 3) Click login button For non-registered users 1) Click the new registration button 2) Fill-up required fields 3) Set password 4) Click register 5) Go back to the login page 6) Type email & password 7) Click login button
	Or simply click on Gmail login (registered and non-registered)
Other Path	Invalid Username or Password

Users can visit and explore products without login but if they want to order something they must have to log in with a valid email and password or Gmail login. Or if they are not registered then they have to complete the registration process then they could complete their orders, make payment, add feedback, etc.

Table 3.2.2: Admin Login Form

Use case for	User login
Actor	Admin (registered)
General Path	1) Type username/email 2) Type password 3) Click login button
	Or simply click on Gmail login (registered and non-registered)
Other Path	Invalid Username or Password

Admin actions are listed below:

- Confirm Orders
- Delete Orders
- Add Products
- Delete Products
- Add another admin

CHAPTER 4

Design Specification

4.1 Front-End Design

4.1.1 Home Page

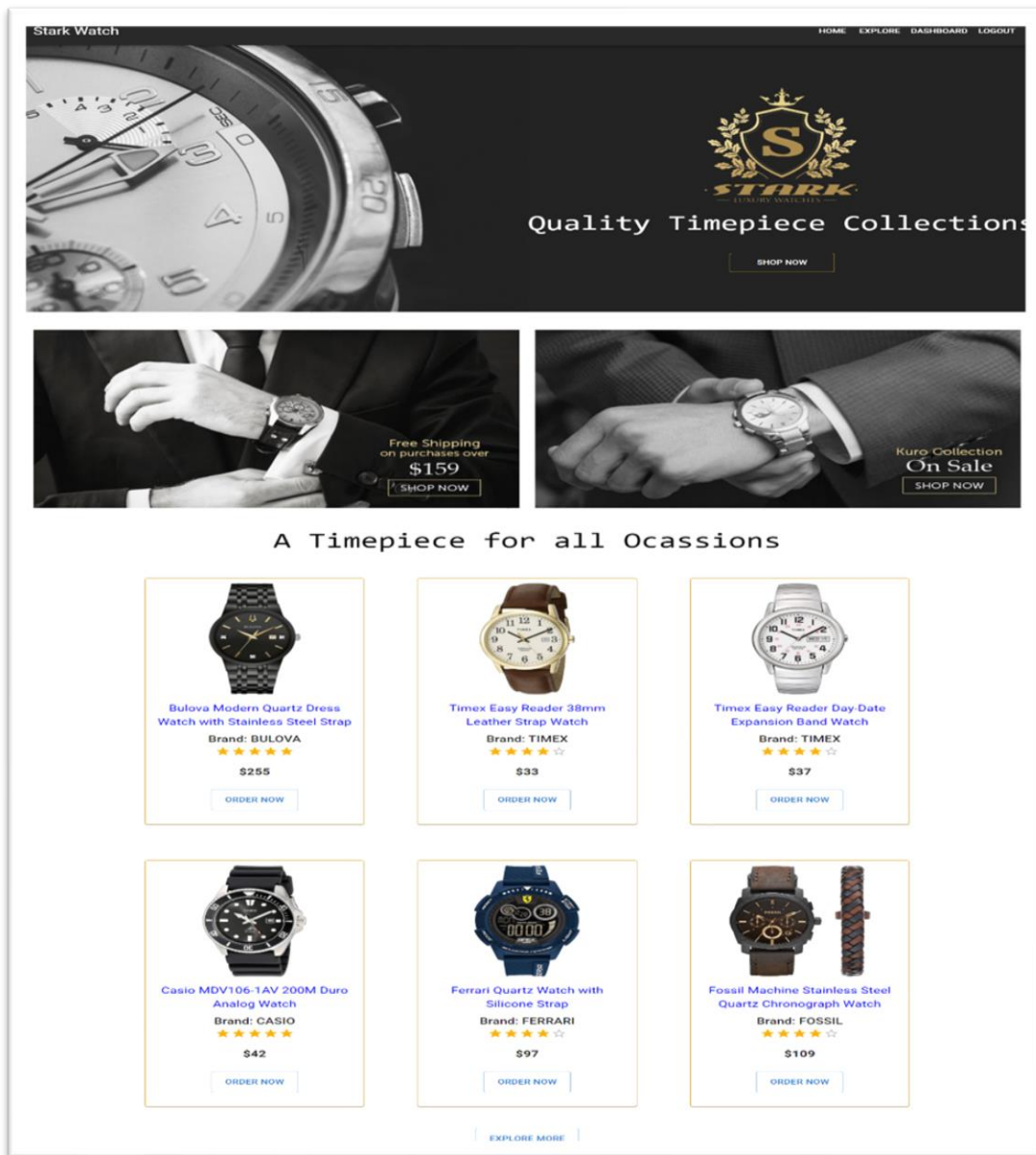


Figure 4.1.1.1: Home Page-1

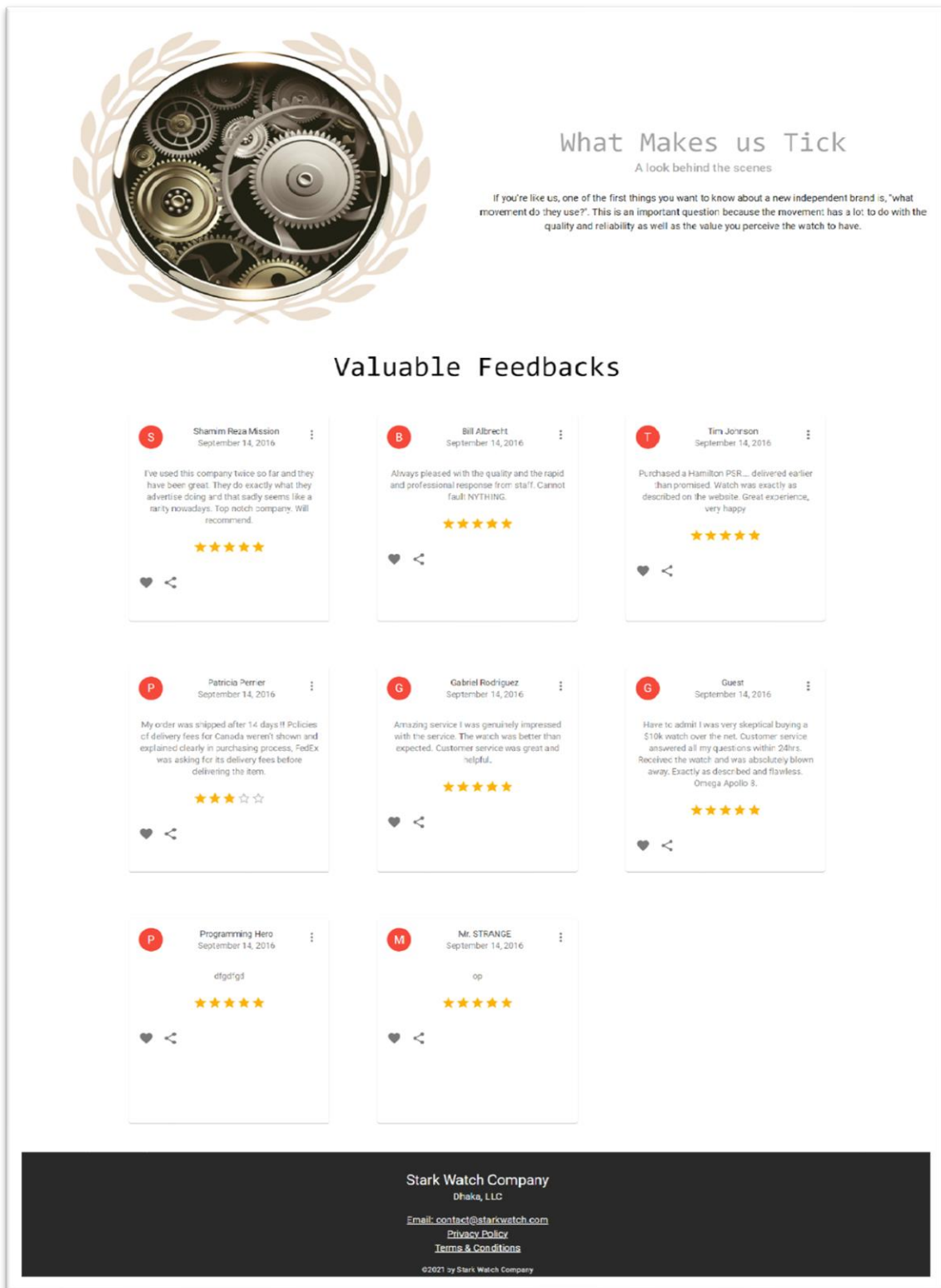


Figure 4.1.1.2: Home Page-2

The Home Page of my web application is shown in **Figure 4.1.1.1** and **Figure 4.1.1.2**. There are 7 parts available. These are:

1. Navbar

- ✓ In Navbar, there are some navigation buttons. It sets as top-fixed position.

2. Top Banner

- ✓ The top Banner section has 2 parts. On left, has a featured image and a company logo and “SHOP NOW” button addressed on the right side.

3. Top Offers

- ✓ Two featured offers are shown in this section.

4. Products

- ✓ Company products are shown there as a grid.

5. Motivation

- ✓ A featured image and some valuable lines have been added to this section.

6. Valuable Feedbacks

- ✓ Users given feedback are shown in this section.

7. Footer

- ✓ It sets as bottom-fixed position.

Users can visit this page without any authentication process. But there are some limitations for them. Limited products are shown on the home page. if they want to explore more they have to go to explore page addressed on the Navbar and below the product section also.

4.1.2 Explore Page

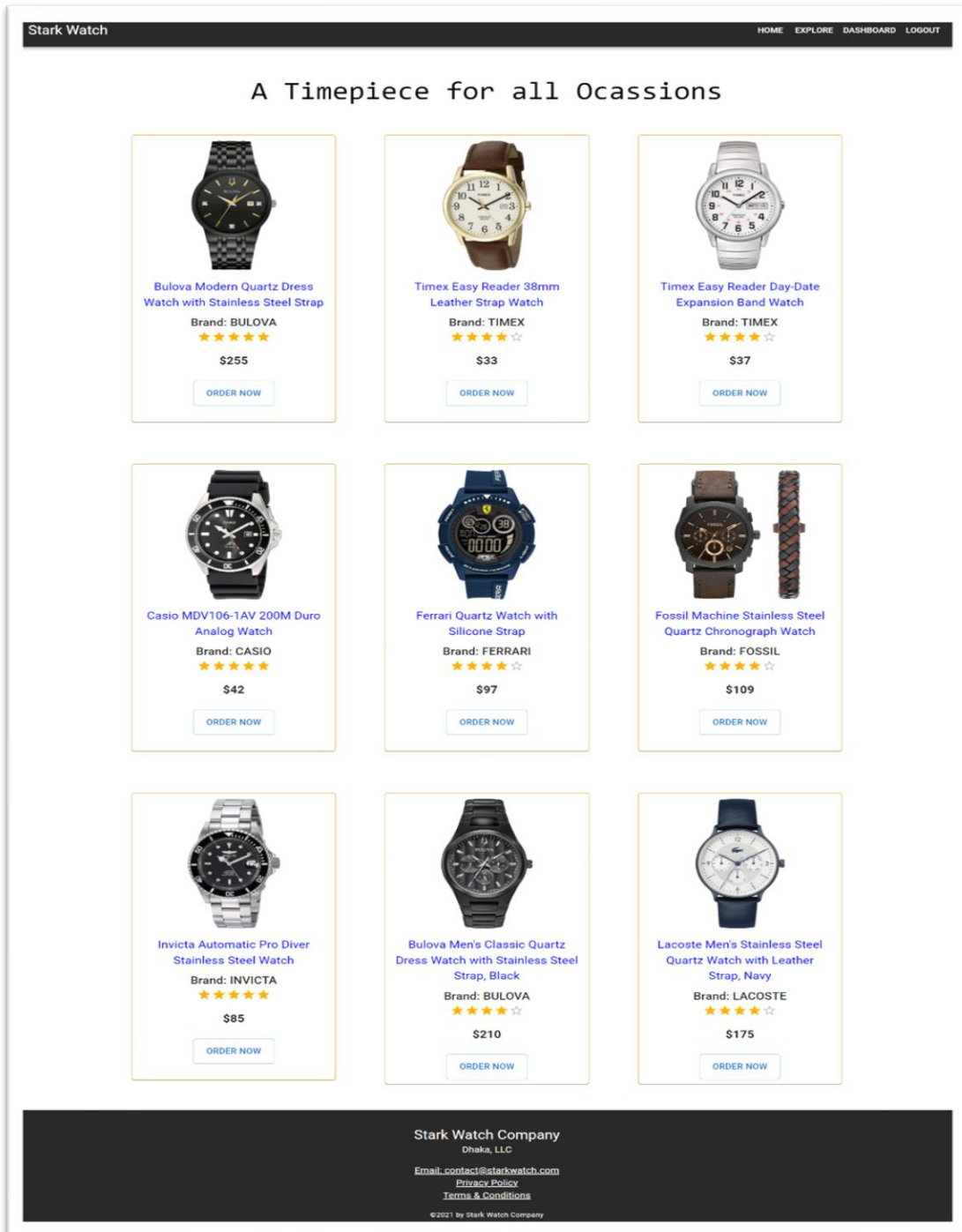


Figure 4.1.2.1: Explore Page

This is the explore page where all products have been shown. Users can visit this without a login. They can see product details, rating, price, etc. If they want to order something then they have to click on the “ORDER NOW” button, given on every product card. If they are already logged in then an order confirmation modal will pop up with some required field. If they are not logged in then a modal will open with a message “Please Login” (Shown in **Figure 4.1.2.2**). There will be a button available named “LOG IN”. After clicking that it will take them directly to the login page.

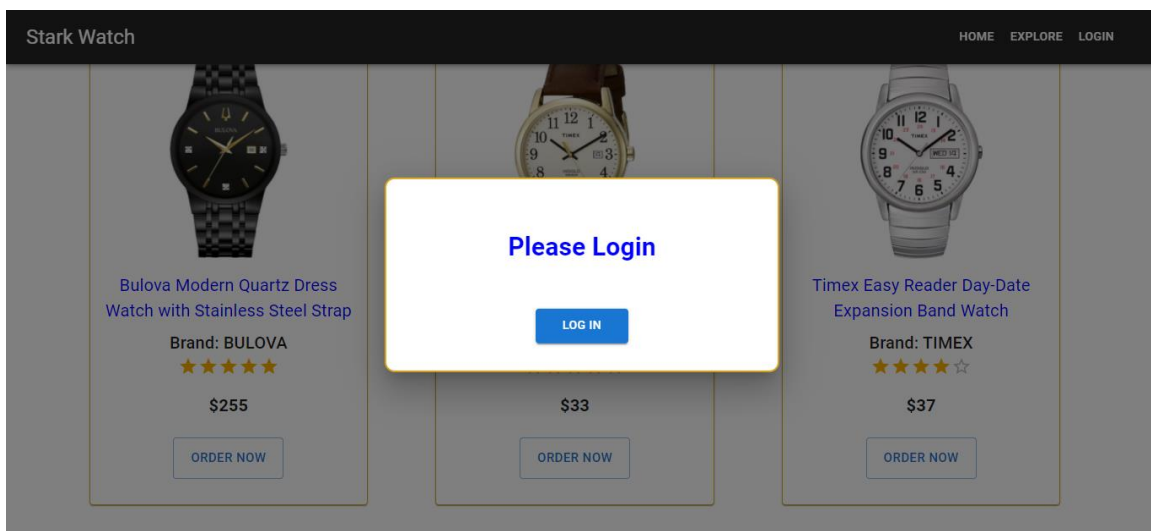


Figure 4.1.2.2: Log in Modal

4.1.3 Registration Page

Stark Watch

HOME EXPLORE LOGIN

Please Register

Your Name *

Your Email *

Your Password *

Re-type your Password *

REGISTER

ALREADY REGISTERED? PLEASE LOGIN

Figure 4.1.3.1: Register Page

This is the registration page of my web application. Users need to register for ordering something from our application. They just need to fill up the form which required a Name, Email, and a valid password to complete the registration process. This process will be handled by Firebase authentication and users' data will be stored in Firebase Database. Or if they are already registered then they can log in from the menu bar where the login button is addressed or the “ALREADY REGISTERED? PLEASE LOGIN” button given below the registration form.

4.1.4 Login Page

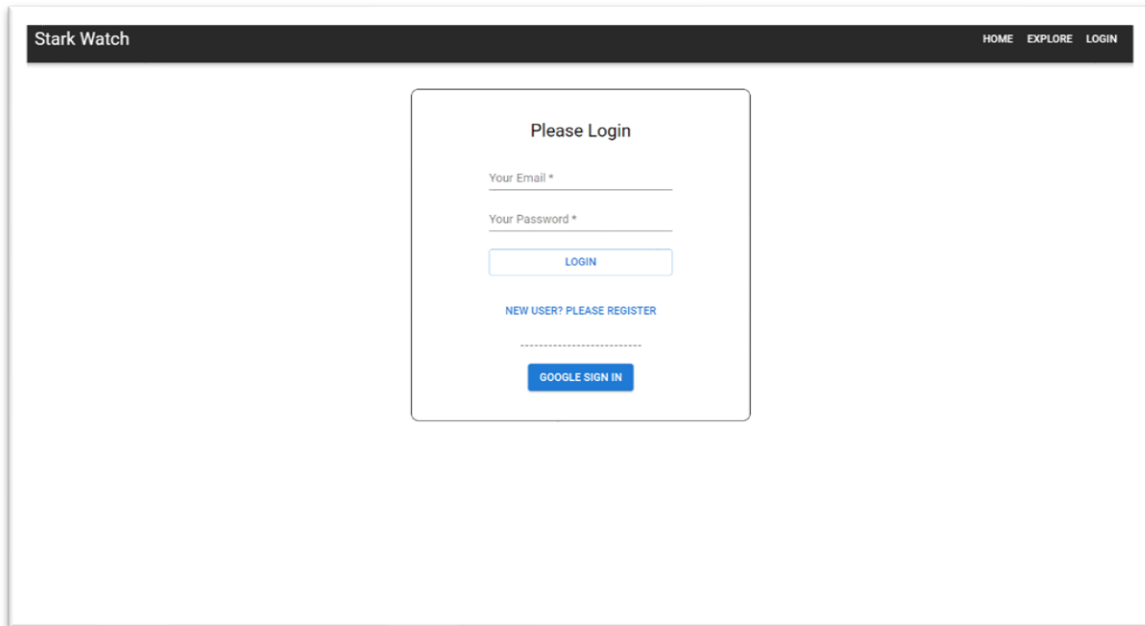
The image shows a web browser window displaying the 'Stark Watch' login page. At the top, a dark navigation bar contains the site name 'Stark Watch' on the left and links for 'HOME', 'EXPLORE', and 'LOGIN' on the right. The main content area is white and features a central login form. The form is titled 'Please Login' and includes two input fields: 'Your Email *' and 'Your Password *'. Below these fields is a light blue 'LOGIN' button. Underneath the button is a link that reads 'NEW USER? PLEASE REGISTER'. At the bottom of the form is a blue 'GOOGLE SIGN IN' button.

Figure 4.1.4.1: Login Page

This is the Login page where a form has been given to log in. Users or Admin can log in by putting their valid Email and Password. This login process will be managed by Firebase Authentication. If they put something invalid then the system will provide an error message and the login process will be terminated.

Instead of using email and password, they can easily sign in by pressing the button called “GOOGLE SIGN IN” given below the Login form. After pressing that a popup will open and will ask for Gmail login.

4.1.5 Order Modal

The screenshot displays the 'Stark Watch' website interface. At the top, a navigation bar includes 'HOME', 'EXPLORE', 'DASHBOARD', and 'LOGOUT'. The main banner reads 'A Timepiece for all Occassions'. Two product cards are visible: a Bulova Modern Quartz Dress Watch with a Stainless Steel Strap (\$255) and a Timex Easy Reader Day-Date Expansion Band Watch (\$37). An order modal is open in the center, titled 'Timex Easy Reader 38mm Leather Strap Watch'. The modal contains a form with the following fields: Name (filled with 'Shamim Reza Mission'), Email (filled with 'iamsrmission@gmail.com'), Phone Number (filled with '+8801796476773'), Address (filled with '221b Baker St, London NW1 6XE, United Kingdom'), and Quantity (filled with '1'). A blue 'SUBMIT' button is at the bottom of the modal.

Figure 4.1.5.1: Order Modal

This is the order modal, this will open when a user will click on the “ORDER NOW” button. As we can see in **Figure 4.15.1**, there is a form with some required fields. When a user wants to order something from our web application, they have to fill up this form. they have to provide their name, valid email, phone number, and quantity of that product that they are interested in. After that, they have to click on Submit button. If their provided information is valid then they will get an order confirmation message on the bottom-left corner of their screen. They can manage their order from Dashboard Section by clicking the “DASHBOARD” button addressed on the menu bar. But this will not be shown until login.

4.1.6 User Dashboard

This dashboard will be available only after successful login. There are several options for users.

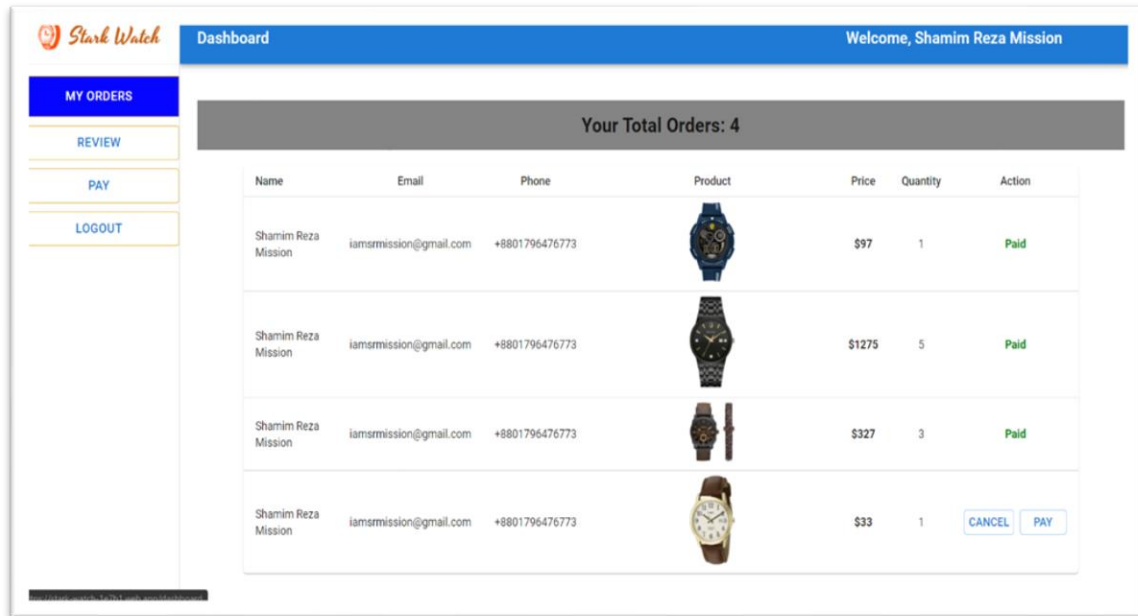
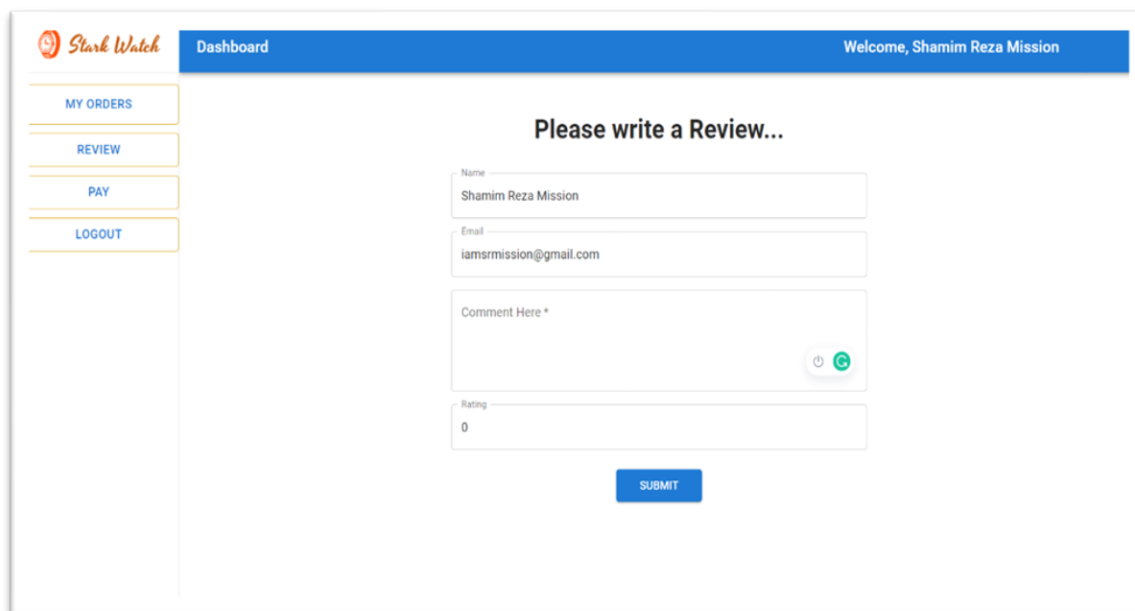


Figure 4.1.6.1: User Dashboard

- **MY ORDERS:** As shown in **Figure 4.1.6.1**, when a user will open this page they can manage their orders. In there they will see their total orders count, their name, email, a phone number that they provided for each order, ordered product image, price, quantity, and Action.

In Action, there will be two-button available named “CANCEL, PAY”. If they want to delete/cancel an order they have to click to cancel button. After pressing that, a windows warning will pop up with a message that “Are you sure to cancel this order” with two more buttons named “Delete, Cancel”. By pressing delete, it will delete that order and will update the orders list instantly. And Pressing cancel will discard the delete operation.

On the other hand, to make a payment they have to press the “PAY” button. That will redirect them to the payment page.



The screenshot shows a web dashboard for 'Stark Watch'. The top navigation bar is blue with the 'Stark Watch' logo on the left, 'Dashboard' in the center, and 'Welcome, Shamim Reza Mission' on the right. A left sidebar contains a vertical menu with buttons: 'MY ORDERS', 'REVIEW', 'PAY', and 'LOGOUT'. The main content area is titled 'Please write a Review...'. It contains a form with the following fields: 'Name' (filled with 'Shamim Reza Mission'), 'Email' (filled with 'iamsrmission@gmail.com'), 'Comment Here *' (a text area with a green speech bubble icon), and 'Rating' (a dropdown menu showing '0'). A blue 'SUBMIT' button is located at the bottom of the form.

Figure 4.1.6.2: Add Review Form

- **REVIEW:** As shown in **Figure 4.1.6.2**, In this review section users can add a review by giving their opinion, input their rating as put a number between 1 to 5. After submission, it will be directly added to the Home Page and the user will get a confirmation message instead.

The screenshot shows a web application interface for 'Stark Watch'. On the left is a sidebar with a vertical menu containing 'MY ORDERS', 'REVIEW', 'PAY', and 'LOGOUT'. The top navigation bar is blue and contains the 'Stark Watch' logo, the word 'Dashboard', and a welcome message 'Welcome, Shamim Reza Mission'. The main content area is titled 'Payment from "Shamim Reza Mission" for "Timex Easy Reader 38mm Leather Strap Watch"'. Below the title, it states 'Payment Amount: \$33'. There is a form for entering payment details, including a card number field with the value '4242 4242 4242 4242', an expiry date field with '04 / 24', and a 'Pay \$33' button.

Figure 4.1.6.3: Payment Form

- **Payment:** As shown in **Figure 4.1.6.3**, On the payment page, users can make a payment by using their international transaction cards. It just required a valid card number, card expiry date, CBC number, and ZIP Code. If they enter all valid information then the payment will proceed with a confirmation message and the Payment button will disable that they can't repay for the same product by mistake.

If their given information is invalid then the payment process will be denied with an error message. The whole payment process will be handled by the "Stripe online payment process" and all important information will be stored in MongoDB.

- **LOGOUT:** By pressing the "LOGOUT" button, the user will be logged out from the web application and will be redirected to Home Page automatically.

4.1.7 Admin Dashboard

When someone will try to log in the system will verify the email, is it the admin's email users? If and only if the email is verified as an admin's then the Admin Dashboard will be available for an Admin. There will be some different options for an Admin which is different from a user's dashboard. Here will be:

Stark Watch

MANAGE ALL ORDERS

ADD A PRODUCT

MAKE ADMIN

MANAGE PRODUCTS

LOGOUT

Dashboard

Welcome, Shamim Reza Mission

Total Orders: 13

Name	Email	Phone	Product	Quantity	Action
Shamim Reza Mission	smission64@gmail.com	+8801796476773	Timex South Street Sport Watch		<a>CANCEL
Shamim Reza Mission	smission64@gmail.com	01701125373	Casio MDV106-1AV 200M Duro Analog Watch		<a>CANCEL
Shamim Reza Mission	alienheart59@gmail.com	01706539587	Casio MDV106-1AV 200M Duro Analog Watch	5	<a>CANCEL
Shamim Reza Mission	alienheart59@gmail.com	+8801796476773	Bulova Modern Quartz Dress Watch with Stainless Steel Strap	4	<a>CANCEL
Programming Hero	admin@admin.com	324456446113121	Timex Easy Reader 38mm Leather Strap Watch		<a>CANCEL
Programming Hero	programminghero006@gmail.com		Timex South Street Sport Watch		<a>CANCEL
Mr. STRANGE	darkprimezahin2004@gmail.com	+8801307141108	Casio MDV106-1AV 200M Duro Analog Watch		<a>CANCEL
Md Rasel Khan	raseletztib@gmail.com	546446	Timex South Street Sport Watch	1	<a>CANCEL
Shamim Reza Mission	iamsmission@gmail.com	+8801796476773	Ferrari Quartz Watch with Silicone Strap	1	<a>CANCEL
Shamim Reza Mission	iamsmission@gmail.com	+8801796476773	Bulova Modern Quartz Dress Watch with Stainless Steel Strap	5	<a>CANCEL
Shamim Reza Mission	iamsmission@gmail.com	+8801796476773	Fossil Machine Stainless Steel Quartz Chronograph Watch	3	<a>CANCEL
Md. Shamim Reza Mission	contact.smission@gmail.com	01701125373	Invicta Automatic Pro Diver Stainless Steel Watch	4	<a>CANCEL
Shamim Reza Mission	iamsmission@gmail.com	+8801796476773	Timex Easy Reader 38mm Leather Strap Watch	1	<a>CANCEL

Figure 4.1.7.1: Manage All Orders Page

- **MANAGE ALL ORDERS:** As shown in Figure 4.1.7.1, in this section an admin can manage all the orders. All users' orders will be listed here. Admin can cancel each order by clicking on the “CANCEL” button.

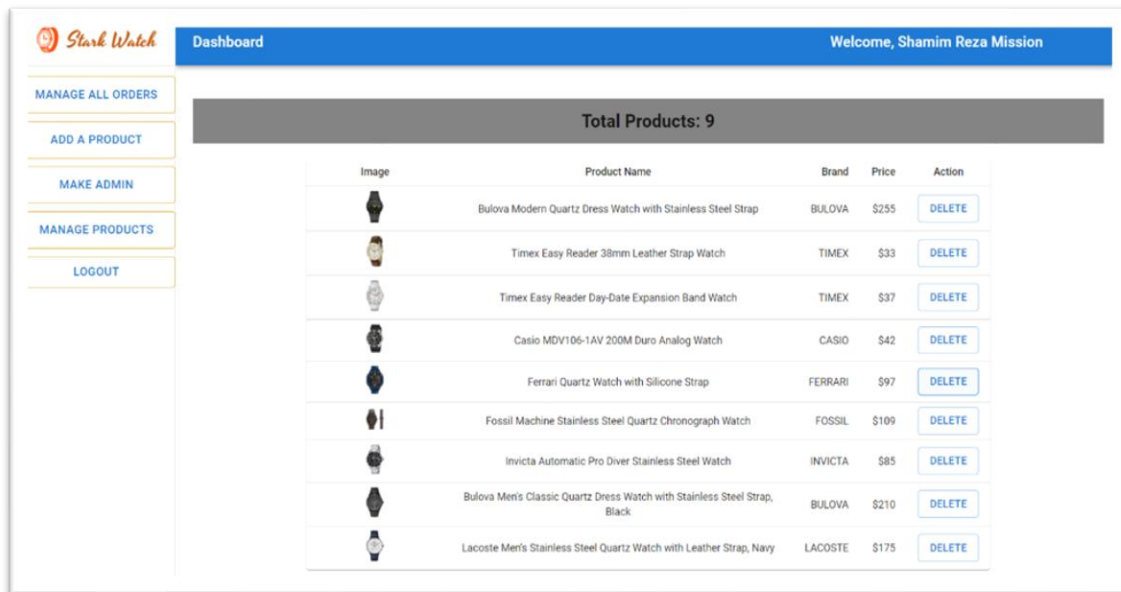


Figure 4.1.7.2: Manage Products Page

- **MANAGE PRODUCTS:** As shown in Figure 4.1.7.2, in this section admin will be able to manage all products available in this web application. Admin can delete any product whenever they want to, by clicking on the “DELETE” button. There is also a double confirmation pop-up option.

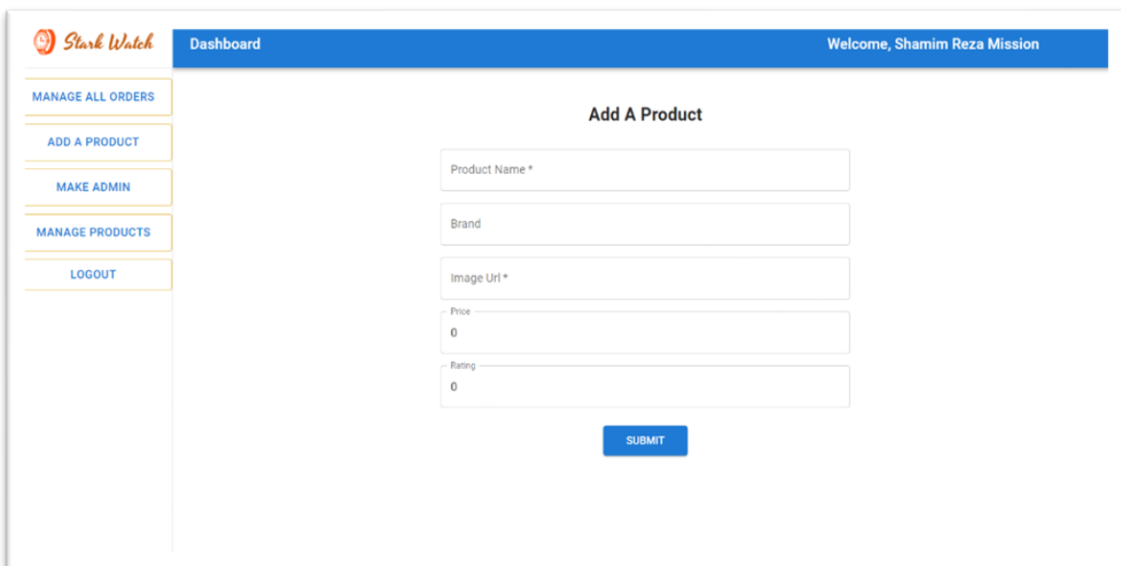
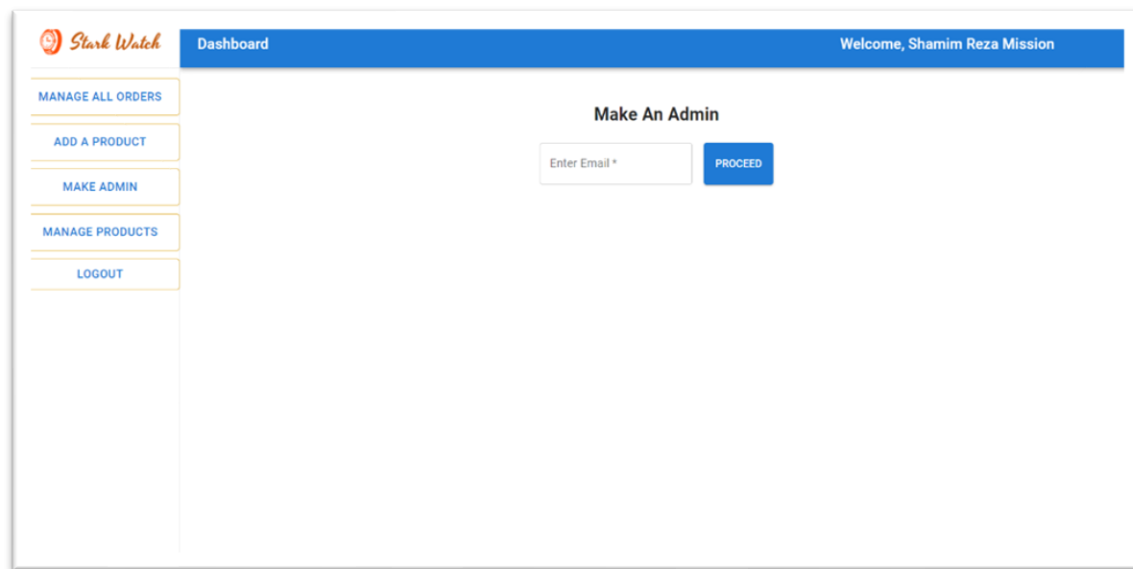


Figure 4.1.7.3: Add a Product Page

- **ADD A PRODUCT:** As shown in **Figure 4.1.7.3**, An Admin can add products from this section without any hassle. Just need to provide the product's information on the form. No need to add something manually to the database. After clicking on the "SUBMIT" button, there will show a confirmation message and The system will automatically fetch it on the database.



The screenshot shows a web application interface for 'Stark Watch'. At the top, there is a blue header bar with the 'Stark Watch' logo on the left, the word 'Dashboard' in the center, and a welcome message 'Welcome, Shamim Reza Mission' on the right. On the left side, there is a vertical sidebar with several menu items: 'MANAGE ALL ORDERS', 'ADD A PRODUCT', 'MAKE ADMIN', 'MANAGE PRODUCTS', and 'LOGOUT'. The main content area is titled 'Make An Admin'. It contains a text input field labeled 'Enter Email *' and a blue button labeled 'PROCEED'.

Figure 4.1.7.4: Add an Admin Page

- **MAKE ADMIN:** As shown in **Figure 4.1.7.4**, There will be an input form and a button named "PROCEED". An admin can add anyone as another admin by putting their email only, and after clicking on the "PROCEED" button, the email will be added to the database and the role will be Admin, and exiting Admin will get a system-generated confirmation message. If something wrong happens, then there will be an error message with a suitable message.

4.2 Back-End Design

4.2.1 MongoDB

I used MongoDB as Database. This is a NoSQL database system. Where data is stored as a document. As shown in **Figure 4.2.1.2**, these are the database collections and stored as documents. There I can check my document size, document count and a lot of options are available here. As shown in **Figure 4.2.1.3**, products data is stored as texts. Manual data added system, edit, update, delete options are available as well. Every data set has a unique `_id`, so there is no chance to mismatch one to another.



Figure 4.2.1.1: MongoDB Login

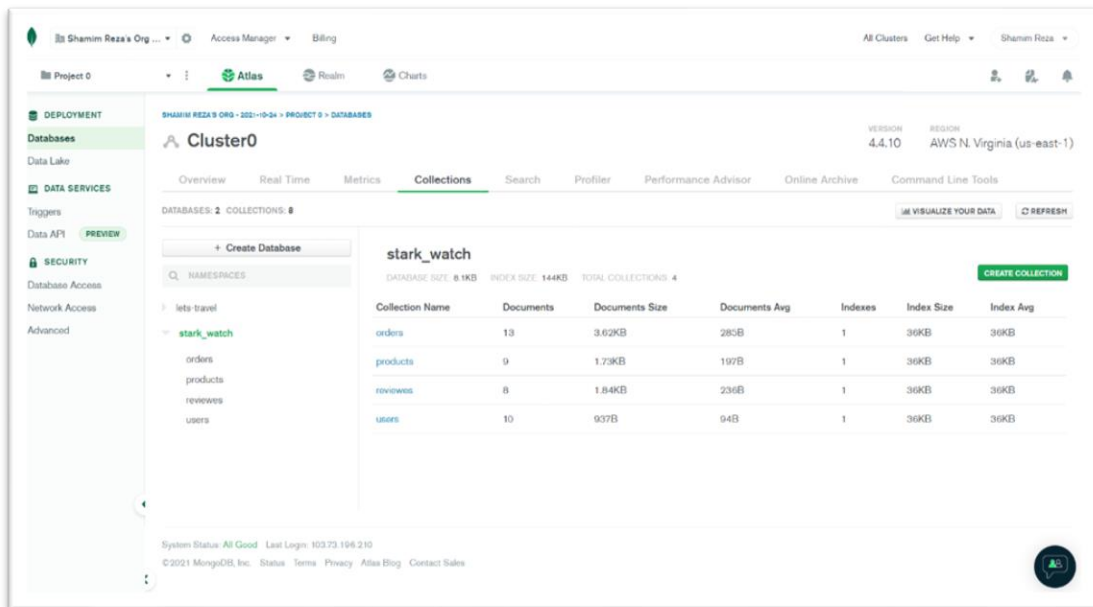


Figure 4.2.1.2: Databases

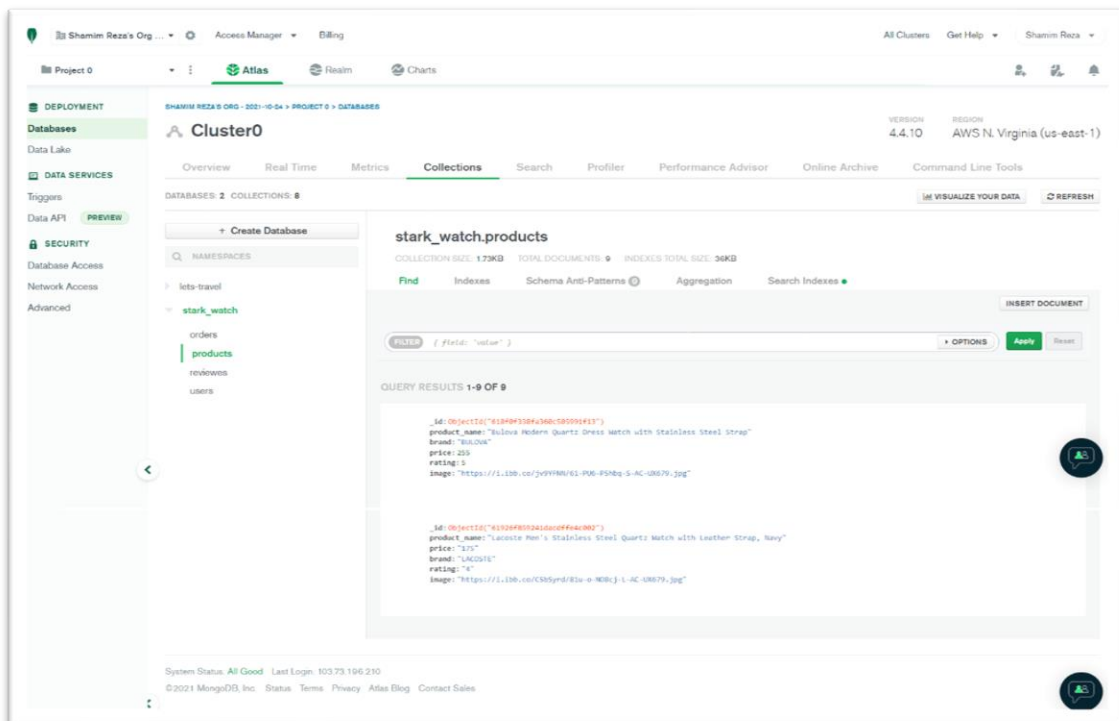


Figure 4.2.1.3: Data Lists

4.2.2 Firebase

I used firebase for users authorization, authentication, and hosting my client-side.

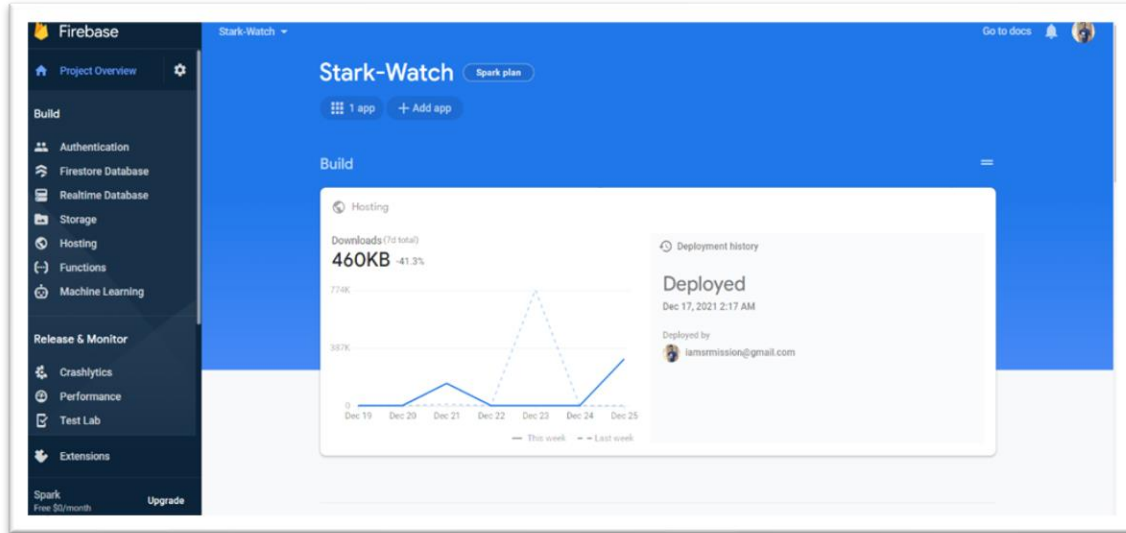


Figure 4.2.2.1: Firebase Console

The screenshot displays the Firebase Authentication console for the 'Stark-Watch' project. The 'Users' tab is selected, showing a list of users. The table includes columns for Identifier, Providers, Created, Signed in, and User UID. The list shows 9 users, with the first few being 'contact.smission@gmail...', 'iamsmission@gmail.com', and 'alien.heart50@gmail.com'.

Identifier	Providers	Created	Signed in	User UID
contact.smission@gmail...	Google	Dec 17, 2021	Dec 17, 2021	ndmsiGRaAsMgN5jCchNl...gpg22
iamsmission@gmail.com	Google	Dec 15, 2021	Dec 27, 2021	16RQkcnDryPfxJ44qZhpOx...Jz2
alien.heart50@gmail.com	Google	Dec 15, 2021	Dec 15, 2021	eFTkDnLBpsWWemD8pRywjzq...
raseizttd@gmail.com	Google	Nov 28, 2021	Nov 28, 2021	eHGTeenNPBYnc2wx41B9HsV8R...
darkprimezahn2004@gma...	Google	Nov 19, 2021	Nov 19, 2021	I2AvicowatRj5qH3DM5NNEadx1
programminghero006@gm...	Google	Nov 18, 2021	Nov 18, 2021	MYMnwOUMHh9S4gKHQV5w3n...
admin@admin.com	Google	Nov 15, 2021	Nov 19, 2021	dSBkcP85PMh9FMVJYSLXBqDF0...
alienheart59@gmail.com	Google	Nov 13, 2021	Dec 8, 2021	IgrYB5eWmsThXUp0o7hHyB4Qh22
smission64@gmail.com	Google	Nov 13, 2021	Dec 27, 2021	YKOCu8ELghJofpPYKzmq2RvX43

Figure 4.2.2.2: Firebase Authentication (User's List)

4.2.3 Heroku

I used Heroku for hosting server-side codes. As shown in **Figure 4.2.3.1**, this is the Heroku dashboard. There I can check server-side deployment status.

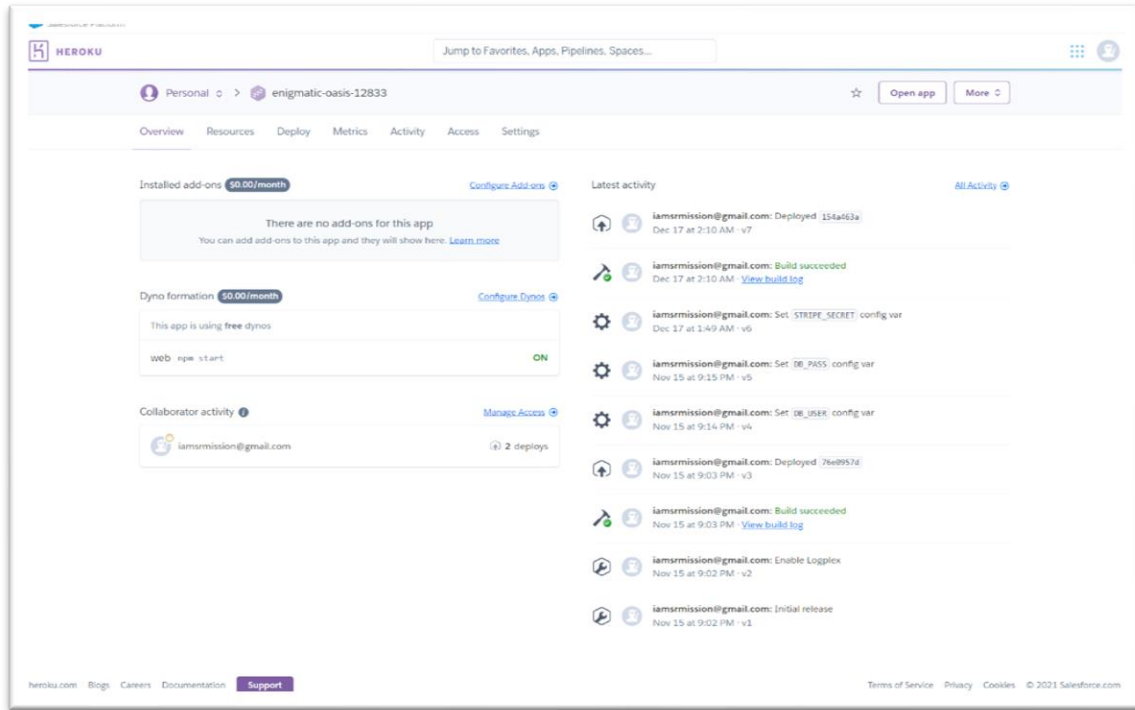


Figure 4.2.3.1: Heroku Dashboard

CHAPTER 5

Conclusion And Future Scope

5.1 Discussion and Conclusion

This project is developed according to the current need in different Fields. This is a Niche Product Web Application that provides a facility for purchasing Niche Product items. To conclude the description of the project. The project, developed using Reacts Js, Material UI, Node.js, Express.js, Firebase, MongoDB, and Heroku is based on the requirement specification of the user and the analysis of the best recent web development language, with flexibility for future enhancement.

5.2 Scope for Further Career

There are huge options to improve in this project and need to add some features. That will be very useful for both users and admin. This project will help me to enhance my knowledge of web development and also help me to take challenges to develop any kind of web application as per market or company or any requirement.

References

[1] MaRS Startup Toolkit, available at << https://learn.marsdd.com/wp-content/uploads/2009/12/Waterfall-product-development-model_20190807.png >>, last accessed on 23-12-2021 at 3:27 AM.

[2] PERFORCE, available at << <https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document> >>, last accessed on 23-12-2021 at 3:40 AM.

[3] HOSTINGER TUTORIALS, available at << <https://www.hostinger.com/tutorials/what-is-html> >>, last accessed on 23-12-2021 at 3:40 AM.

[4] CSS - Wikipedia, available at << <https://en.wikipedia.org/wiki/CSS> >>, last accessed on 23-12-2021 at 3:40 AM.

[5] What is JavaScript? available at << https://deepak379mandal.medium.com/what-is-javascript-d18834291fd0?source=post_internal_links >>, last accessed on 23-12-2021 at 3:40 AM.

[6] React - Wikipedia available at << [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)) >>, last accessed on 23-12-2021 at 3:40 AM.

[7] How to Implement Material-UI in React available at << <https://www.section.io/engineering-education/how-to-implement-material-ui-in-react> >>, last accessed on 23-12-2021 at 3:40 AM.

[8] What is Node.js? available at << <https://www.tutorialsteacher.com/nodejs/what-is-nodejs> >>, last accessed on 23-12-2021 at 3:40 AM.

[9] Express.js - Wikipedia available at << <https://en.wikipedia.org/wiki/Express.js> >>, last accessed on 23-12-2021 at 3:40 AM.

[10] What is Firebase? | VentureBeat available at << <https://venturebeat.com/2021/08/02/what-is-firebase> >>, last accessed on 23-12-2021 at 3:40 AM.

[11] What is MongoDB? A definition from WhatIs.com available at << <https://searchdatamanagement.techtarget.com/definition/MongoDB> >>, last accessed on 23-12-2021 at 3:40 AM.

[12] Heroku - Wikipedia available at << <https://en.wikipedia.org/wiki/Heroku> >>, last accessed on 23-12-2021 at 3:40 AM.

Web Application

ORIGINALITY REPORT

23%

SIMILARITY INDEX

%

INTERNET SOURCES

%

PUBLICATIONS

23%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Daffodil International University

Student Paper

3%

2

Submitted to Griffith College Dublin

Student Paper

3%

3

Submitted to Technological Institute of the Philippines

Student Paper

3%

4

Submitted to Lovely Professional University

Student Paper

2%

5

Submitted to DIT university

Student Paper

2%

6

Submitted to Université Saint-Esprit Kaslik

Student Paper

1%

7

Submitted to Anadolu University

Student Paper

1%

8

Submitted to Universiti Brunei Darussalam

Student Paper

1%

9

Submitted to Universiti Teknologi Petronas

Student Paper

1%

10	Submitted to University of Wolverhampton Student Paper	1 %
11	Submitted to RMIT University Student Paper	1 %
12	Submitted to University of Greenwich Student Paper	1 %
13	Submitted to University of Mauritius Student Paper	1 %
14	Submitted to Rowan University Student Paper	1 %
15	Submitted to Luton Sixth Form College, Bedfordshire Student Paper	<1 %
16	Submitted to National College of Ireland Student Paper	<1 %
17	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
18	Submitted to University of Hertfordshire Student Paper	<1 %
19	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
20	Submitted to Colorado Technical University Student Paper	<1 %
21	Submitted to Higher Education Commission Pakistan Student Paper	<1 %