**JARVIS E-COMMERCE WEBSITE**

A Project Report

*Submitted by*

**GOKULPRASANTH S (35523U18020)**

*in partial fulfillment for the award of the degree*

**BACHELOR OF SCIENCE**

# IN

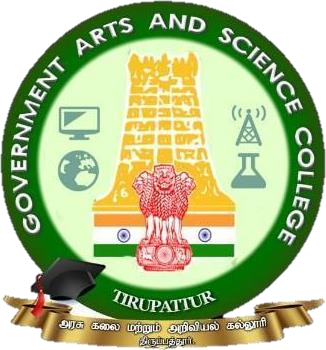
**COMPUTER SCIENCE**

*Under the Supervision of*

**Dr.A.PRIYA, M.Sc., M.Phil., Ph.D.,**

Head & Assistant Professor

Department of Computer Science



**GOVERNMENTARTS AND SCIENCE COLLEGE,**

**Tirupattur – 635901**

**DEPARTMENT OF COMPUTER SCIENCE**

**OCTOBER 2025**

# CANDIDATE’S DECLARATION

I hereby certify that the project entitled **“JARVIS E-COMMERCE WEBSITE”** submitted by **GOKULPRASANTH S (35523U18020**

in partial fulfillment of the requirement for the award of degree of the **B Sc. (COMPUTER SCIENCE)** submitted at **GOVERNMENT ARTS AND SCIENCE COLLEGE,**

**TIRUPATTUR** is an authentic record of my own work carried out during a period from **2025** to **2026** under the guidance of **Dr. A. PRIYA, HEAD &**

**ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE**.

Signature of the Students

1.

Place:

Date:

# CERTIFICATE

This is to certify that the project titled “**JARVIS E-COMMERCE WEBSITE**” is the bonafide work carried out by **GOKULPRASANTH S (35523U18020**in partial fulfillment of the requirement for the award of degree of the **B.SC.**

**(COMPUTER SCIENCE)** submitted at **GOVERNMENT ARTS AND**

**SCIENCE COLLEGE, TIRUPATTUR** is an authentic record his/her work carried out during a period from 2025 to 2026 under the guidance of **Dr.**

**A.PRIYA, HEAD & ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE**.

The Major Project Viva-Voce Examination has been held on .

Signature of the Guide Signature of the HOD

Internal Examiner External

Examiner

# ACKNOWLEDGEMENT

It gives as a great pleasure and satisfaction in presenting this report on my Project work as a part of partial fulfillment for the Bachelor of Science in Computer Science in sincere gratitude to several people, it would have been impossible for me to carry out the desired work.

I would like to express our sincere thanks to **Dr. P. KUMARESHAN, M.Sc., M.Phil., Ph.D.,** Principal, for giving me the opportunity to complete my project in this Institution.

I would like to express my heartfelt thanks to **Dr. A. PRIYA, M.Sc., M.Phil., Ph.D., Head of the Department, Department of Computer Science** for her constant support and Encouragement during this phase and forgiving us valuable suggestions time to time.

I would like to acknowledge and give my warmest thanks to my supervisor

**Dr. A.Priya, M.Sc., M.Phil., Ph.D., Head & Assistant Professor, Department of Computer Science** who made this work possible. His guidance and advice carried me through all the stages of writing my project.

I would also like to give special thanks to my **family** as a whole for their continuous support and understanding when undertaking my project and writing my project. Your prayer for me was what sustained me this far.

Finally, I would like to thank God, for letting me through all the difficulties. I have experienced your guidance day by day. You are the one who let me finish my degree. I will keep on trusting you for my future.

**THIRUVALLUVAR UNIVERSITY**

College Name **: Government Arts and Science College,**

**Tirupattur – 635 901.**

Course **: B.Sc. Computer Science**

Student Name **: GOKULPRASANTH S**

Register Number **: 35523U18020**

Title of the project **: JARVIS E- COMMERCE WEB APPLICATION**

Name Of The Internal Guide **: Dr. A. PRIYA**

Qualification **: M.Sc., M.Phil., Ph.D.,**

Teaching Experience **: 25 Years**

Place **: Tirupattur**

Date **:**

Signature of Internal Guide

Name of the HOD **: Dr. A. PRIYA, M.Sc., M.Phil., Ph.D.,**

Designation **: Head of the Department,**

**Department of Computer Science**

Place **: Tirupattur**

Date **:**

Signature of the HOD

# ABSTRACT

This Project is a Business to Consumer based E-commerce shopping Website which developed using JAVASCRIPT and NOSQL. The project objective is to deliver the online shopping.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. The primary goal of an e-commerce site is to sell goods and services online. This project is a web-based shopping system for an existing shop. The project objective is to deliver the online shopping application. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using a web site. Thus the customer will get the service of online shopping and home delivery from this shop. E-commerce is mainly useful for haven’t time to go shopping or for comfortably to the customers. Those are just entered into this website and bought they want at any time they can visit the web-site.

Customer will get their items just sitting at home.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as flipkart or eBay. Since the application is available in the Smartphone it is easily accessible and always available.

From this project, Firebase can handle the details of customer, stock, sales, shop billing and login details.

|  |  |  |
| --- | --- | --- |
|  | **INDEX** |  |
| **CHAPTER NO.** | **CONTENTS** | **PAGE NO.** |
| **1** | INTRODUCTION |  |
| **1.1** | OVERVIEW |  |
| **2** | SYSTEM REQURIREMENT |  |
| **2.1** | SYSTEM ANALYSIS |  |
| **2.3** | SYSTEM DESIGN SPECIFICATION |  |
| **3** | DATABASE DESIGN |  |
| **3.1** | USE CASE DIAGRAM |  |
| **3.2** | ER DIAGRAM |  |
| **3.3** | DATA NORMALIZATION |  |
| **4** | SOFTWARE TESTING |  |
| **4.1** | SYSTEM TESTING |  |
| **4.2** | WEBSITE SCREENSHOTS |  |
| **4.3** | DATABASE DESIGN |  |
| **5** | FRONTEND CODING |  |
| **5.1** | BACKEND CODING |  |
| **6** | FUTURE SCOPE |  |
| **7** | CONCLUSION |  |

**CHAPTER I**

**INTRODUCTION**

# INTRODUCTION

Electronic commerce, commonly known as E-commerce, is a trading in products or services using network, such as Internet. Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange, inventory management system and automate data collection systems.

Modern electronic commerce typically uses the World Wide Web for at least one part of the transaction life cycle, although it may also use other technologies such as e-mail.

E- Commerce business some or All of the following:

* Online shopping web site for retail sale direct to consumer
* Providing or participating in online marketplace, which process third party business to customer and customer to business sales
* Marketing to prospective and established customer by e-mail or by fax.
* Engaging in retail for launching new products and services

India has always been a land of great potential. The socioeconomic condition of the country has improved many folds and now India is emerging as one of the leading countries in the world. Moreover, with the population of 100 crore and growth a growth it can be compared to marketing giant. Hence it can be well judged in India why online shopping in India is growing at such rate. As technologies are spreading in in the remote villages and many Job opportunity are presenting themselves as to the youth and more people are gaining awareness and customer is growing one commerce site. It also provides an opportunity for the large and small seller to come forward and sell their products on the online market by providing them space to sell their goods at a reasonable price.

# OVERVIEW

The project entitled **JARVIS E-COMMERCE ONLINE SHOPPING WEBSITE** enables customer to buy mobiles or accessories from anywhere through online. This web application advertises some of the products for shopping. To buy products, customer has to create an account. Those who does not have an account, they can only view the available product. They can’t buy it. Once the customer has created account, not only he can view the products, he can also add the product to the cart and also, he can place an order to buy those products. This application then generates bill for that particular customer. After the confirmation, the customer has to enter his credit card details to buy those products.

1. **Product Perspective**

The product will be developed completely independent and dynamic website. Customer must have an account to purchase the product. This application stores all the information in the database which can be retrieved whenever needed and all the validations are performed during the entry of the data by the user thus ensuring that the user cannot enter any wrong data which could cause problem later.

1. **Product Function**

Initially customer has to register to the website to access most of the features of the application. The customer has to enter the details like username, password…etc. After registration customer will be able to purchase products and the purchased product can be added to the cart. Later customer has to enter his credit card details to buy the products.

1. **User Characteristics:** The user of this product is supposed to be fairly educated about the usage of the computers. He should understand how to store products and he should have knowledge about various products so that they could be saved. A person who has no knowledge of computers will find it difficult to understand the system. But with a little knowledge it will be very easy to handle the project.

1. **Admin:** Admin adds the new product and accessories and stores in the database which can be retrieved and used whenever needed and all the validation are performed during the entry of the data. Thus, it ensures that the user cannot enter any wrong data which would cause problem later.
2. **User:** This application allows the user to access all the products available. To buy the products, customer must create an account in this website.
3. **General constraints:** The constraints of this project are - system must support the runtime files of visual studio 2008 and must be able to run all the web pages.
4. **Assumption and dependencies:** The project depend on the user’s ability to understand the features of the online shopping and able to use the best of it. If the internet connection is not proper then this application will not work.

**3. Functional requirements:**

**Login Module:**

Input: Admin enters the Login Id and password.

Process Definition: Check’s login Id and password is valid or not.

Output: Admin is directed to next page where he can add, delete or update the products.

**New user Module:**

Input: Customer enters their details to create new account.

Process Definition: Checks whether he entered all the details or not.

Output: Customer is directed to the next page where he can view all the available products

**Product Module:**

Input: Admin adds the new product which can be viewed by the users. Process Definition: Checks whether all the fields are entered properly or not. Output: Records will be added to the database.

**Search Module:**

Input: This module helps the customer to ease his search based on his interest. The search can be done on different categories like mobile model name, model number, color, price etc

Process Definition: It retrieves the selected category from the database

Output: Displays what the subscribers like to search.

**Cart Module:**

Input: User can select any number of Mobile and add to the cart. He can also remove from the cart if he dislikes it later.

Process Definition: Checks whether all the fields are entered properly or not. Output: Records will be added to the database.

**Payment Module:**

Input: This module describes the payment done by the customer. The payment information can include information like the model purchased, quantity, mode of payment etc. Process Definition: Checks whether all the fields are entered or not. Output: Displays the payment done by the customer.

**Objective of the project:**

This website helps customer to find different products, their features, and new updates easily. It is designed such a way that one can view all the updates of the products from any place through online. The software will help in easy maintaining and updating products in the website for the administrator. Also, quick and easy comparison of different products for the customers.

**Scope of the project:**

This system will reduce the manual operation required to maintain all the records of booking information. And also generates the various reports for analysis. Main concept of the project is to enter transaction reports and to maintain customer records. Hence this software can be used in any mobile showroom to maintain their record easily.

# CHAPTER II

**REQUIREMENTS**

## SYSTEM REQUIREMENTS

### Hardware Requirements

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important

|  |  |
| --- | --- |
| **Processor** | Intel CORE i3 |
| **RAM** | 8.0 GB |
| **Hard Disk Drive** | Space pace require Upto 32 GB |

### Software Requirement

|  |  |
| --- | --- |
| **Number** | **Description** |
| **1** | WINDOWS 8, 10, 11 |
| **2** | HTML/ CSS/ JAVASCRIPT |
| **3** | GITHUB SERVER |
| **4** | FIREBASE STUDIO |
| **5** | NOSQL |

**Language Used:**

|  |  |
| --- | --- |
| Front End: | HTML, CSS, BOOTSTRAP,  JavaScript |
| Back End: | Firebase, MySQL |
| Local host: | GitHub Server |

## SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

1. **Existing system**

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item.

* + User must go to shop and select products.

* + It is difficult to identify the required product.

* + Description of the product is limited.

* + It is a time-consuming process.

* + Not in reach of distant users.

1. **Proposed system**

In the proposed system customer need not go to the shop for buying the products. He can order the product he wish to buy through the website in his Smartphone. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

1. **Feasibility study**

Feasibility requirements are undertaken to have a view of understanding the features that are incorporated in terms of the requirements to design and implementation. The references of the feasibility study will be divided into multiple sections so that the understandability can be Organized so for example if we have to design software the required investment and flow of cash has to be properly analyzed in the same way technical associations are acquired to be discussed. a. Technical feasibility.

1. Operational feasibility.

1. Economic feasibility.

1. **Technical feasibility**

The customization that has to be operational when a particular identity or field is changed has to be replicated in real time which has to be properly designed and tested for the workability. All types of working references are provided in terms of the resources within the company premises. The integrated settings that are required for the accompanied sharing and alert system will also be associated properly and the vendor integrations will be tested for security provisions. The accessibility of the standards that are required will be properly define so as to acknowledge the formats of reports which has to be in-clubbed in different types of reporting sections that are incorporated.

1. **Operational feasibility**

Apprehension of working is required to be properly planed as when the system will be provided to multiple users a proper support has to be provided as multiple integrated functionalities are provided. The web applications that may arise has to be drafted it so that in the real time working it can be properly acknowledged as multiple Global finance will be incorporated to the system. A proper documentation will be provided with all variations of report and the standing in the customization options that are provided under the setting

1. **Financial feasibility**

Financial return on investment calculation so will be done by the financial team with the help of multiple associated software that are available will be used. Considerations of the cash flow will be determined as the source of income has to be properly acknowledged

## SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design. Specifications to performance specification. System design has two phases of development

1. Logical design

1. Physical design

During logical design phase the analyst describes inputs (sources), output s(destinations), databases (data sores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources.

Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

## INPUT AND OUTPUT DESIGN

1. **INPUT DESIGN:**

Input design is the link that ties the information system into the world of its users.

The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design.

Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

1. **OUTPUT DESIGN:**

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

1. **DATABASE DESIGN:**

Databases are the storehouses of data used in the software systems. The data is stored in JSON format the database. Several files are created for the manipulation of the data for the system.

Two essential settings for a database are

* the field that is unique for all the record occurrences.

* the field used to set relation between tables.

* Normalization is a technique to avoid redundancy in the tables

## FRONT END

**HTML, CSS, JAVA SCRIPT** are utilized to implement the frontend.



**HTML**

Every webpage you look at is written in a language called HTML. You can think of HTML as the skeleton that gives every webpage structure. In this course, we'll use HTML to add paragraphs, headings, images and links to a webpage. In the editor to the right, there's a tab called test.html. This is the file we'll type our HTML into. Like any language, it has its own special syntax. A browser's job is to transform the code in test.html into a recognizable webpage! It knows how to lay out the page by following the HTML syntax.



**CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.[1] Most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content.



**JAVA SCRIPT**

JS is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously,and alter the document content that is displayed.

Java Script is used to create pop-up windows displaying different alerts in the system like

“User registered successfully”,” Product added to cart” etc.

Html, Css, Java Script Relationship:



## CHAPTER III

### DATABASE DESIGN

**DATABASE DESIGN**

**DATA FLOW DIAGRAM**

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So, it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information oriented or a process-oriented system flow-chart. For these reasons DFD’s are often referred to as logical data flow diagrams.

**EXTERNAL ENTITY**

An external entity is a source or destination of a data flow. Only those entities which originate or receive data are represented on a data flow diagram. The symbol used is a rectangular box.

#### Entity

**PROCESS**

A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.

#### Process

**DATAFLOW**

The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a processand may be written, verbal or electronic. Each data flow may be referenced by the processes or data stores at its head and tail, or by a description of its contents.

#### Data Flow



**DATA STORE**

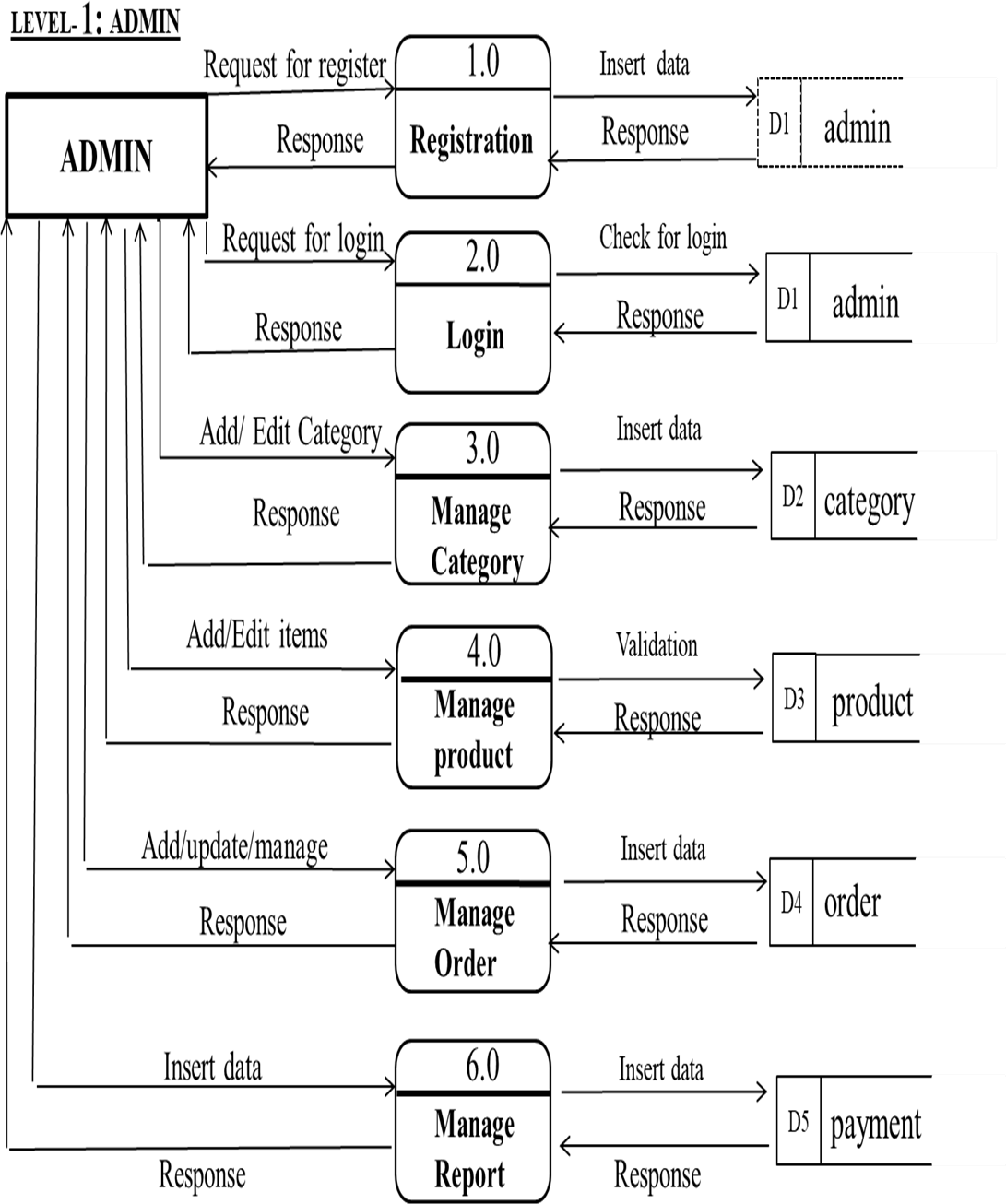
A data store is a holding place for information within the system: It is represented by an open- ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example, batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

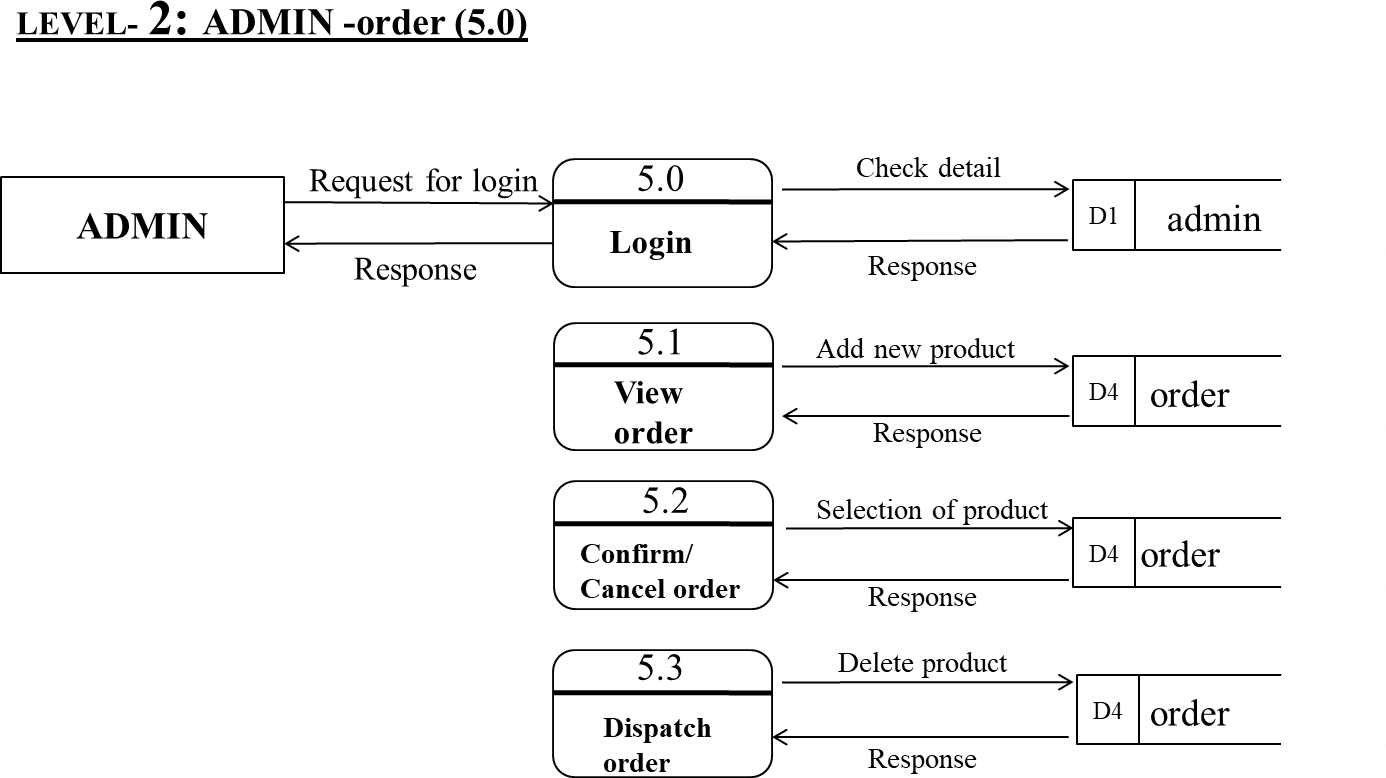
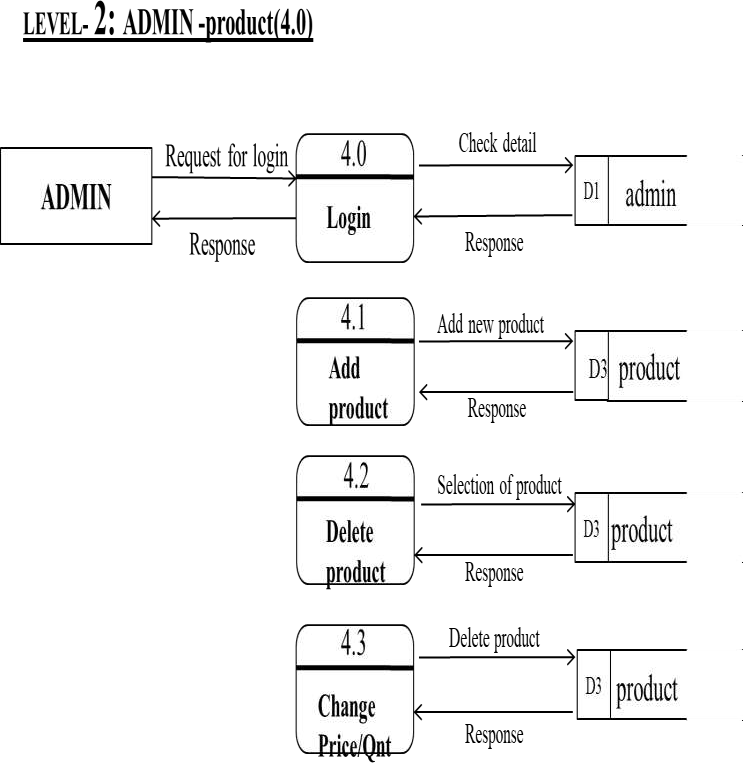
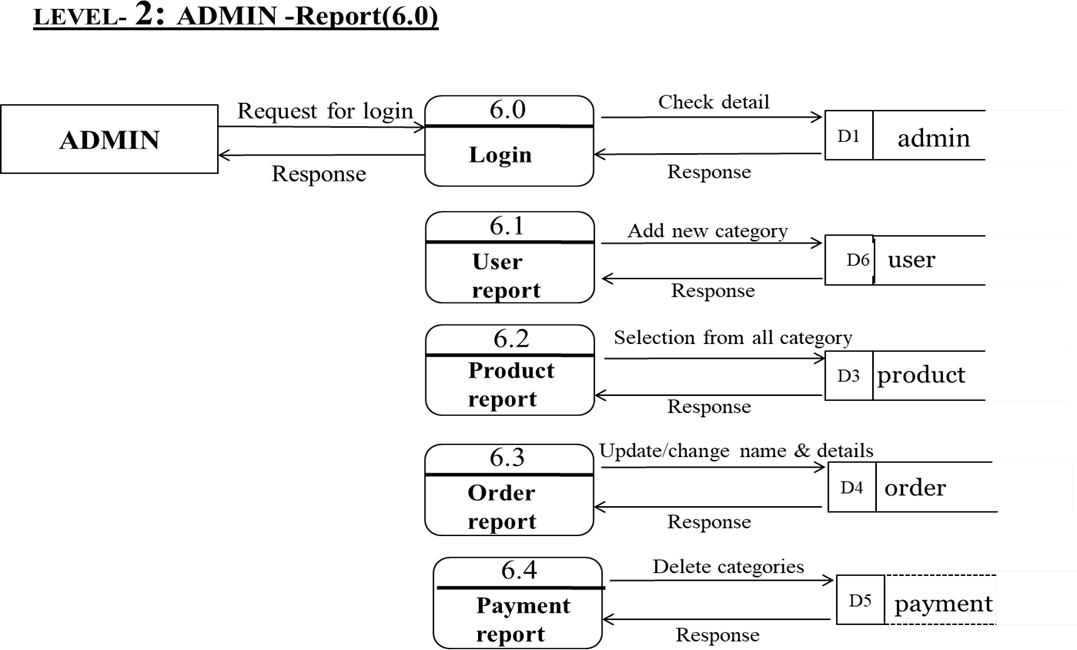
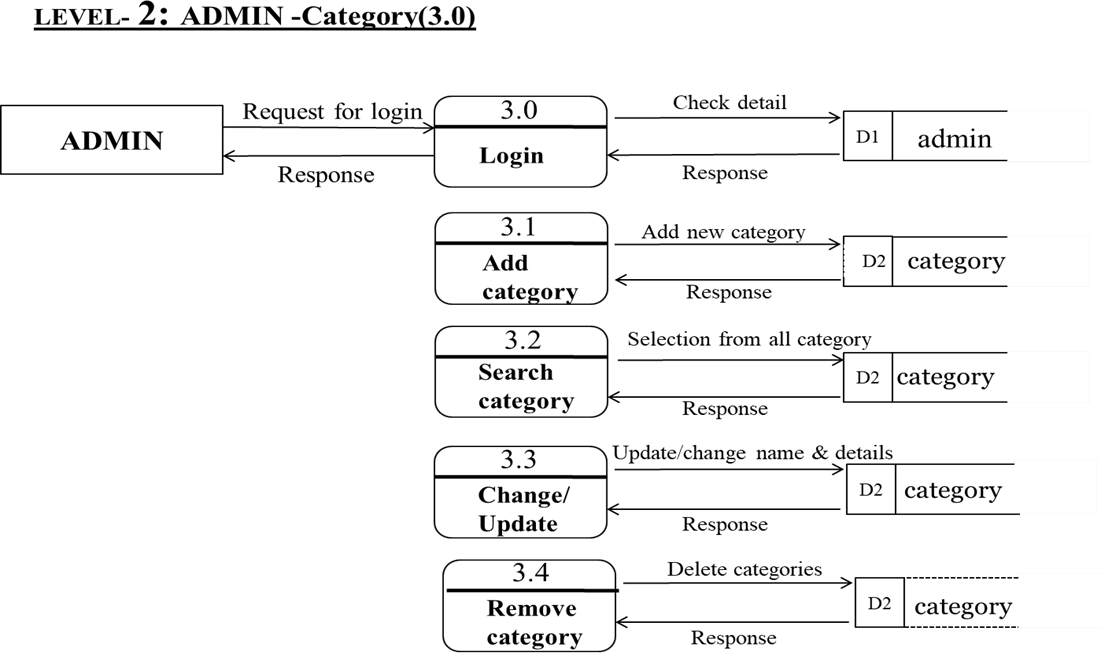
**Data store**

#### Context level Data Flow Diagram



**Admin. Data Flow Diagram level 1**

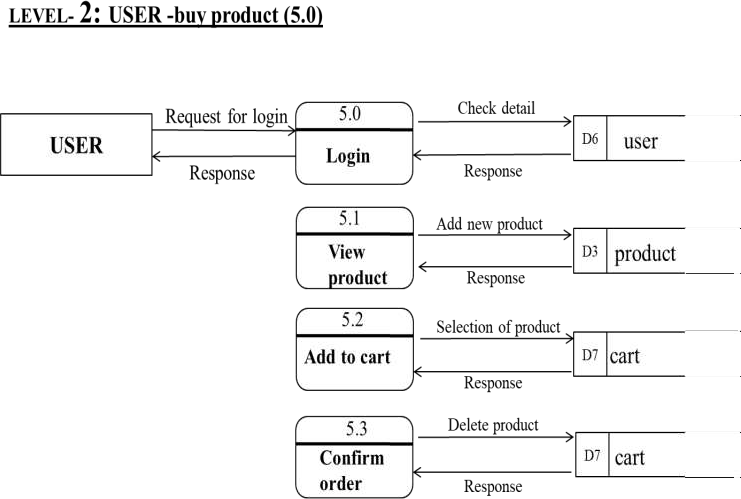
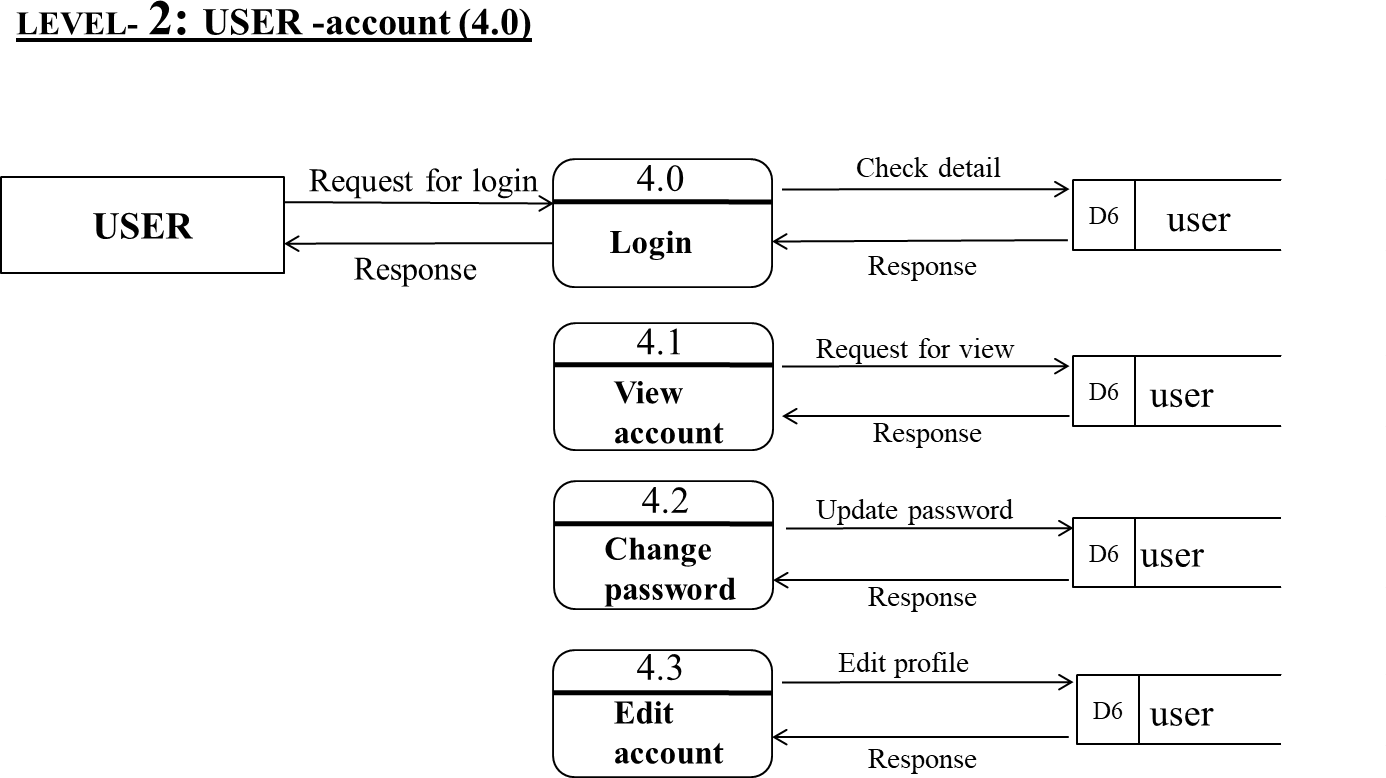
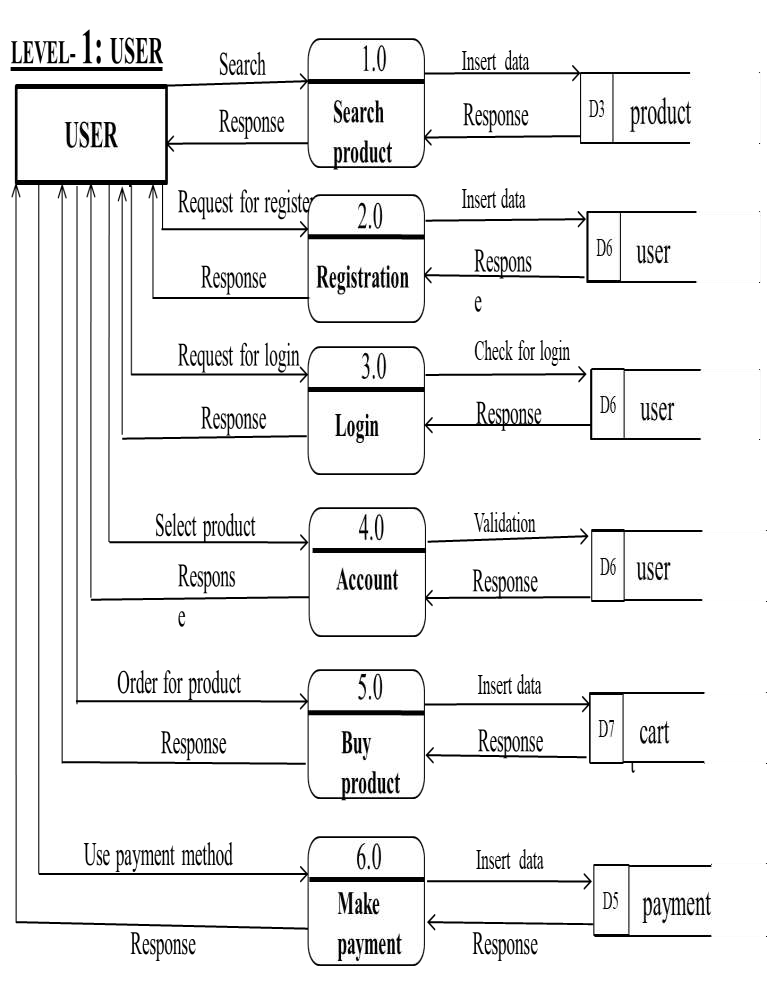




**Data Flow Diagram level 1 USER**

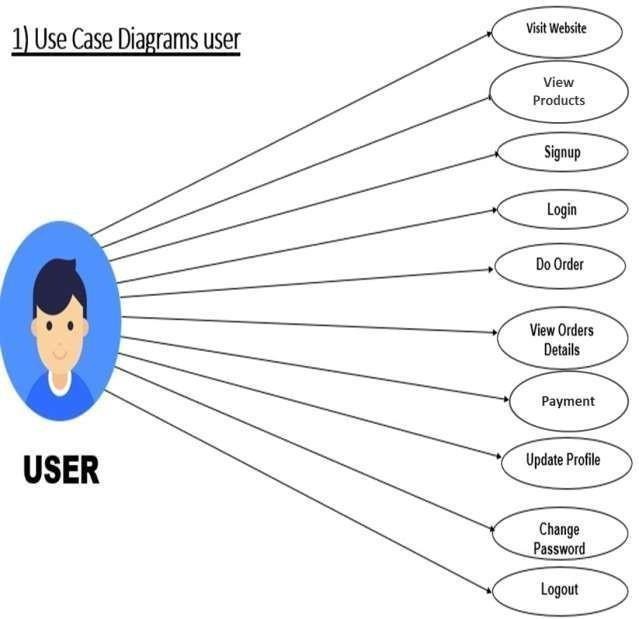
**fig**

**3.0**



#### USE CASE DAIGRAM

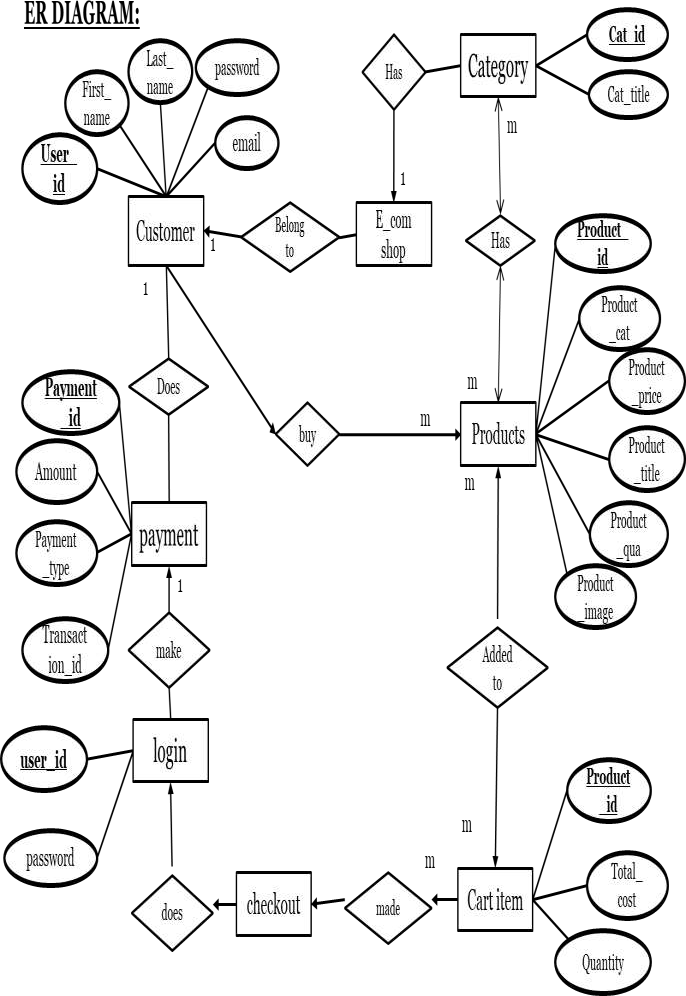
A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



2



**Entity Relationship**



# CHAPTER IV

**TESTING**

## Data Normalization

Normalization is a database design technique that reduces data redundancy and eliminates Undesirable characteristics like insertion Update and Deletion anomalies. Normalization rules divides larger table into smaller tables and links then using relationship. The Purpose of

Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored legally



## Software Testing

Software Testing is the process of identifying the security, correctness, completeness and quality of the developed computer software. Testing is a process of technical searching/investigation, performed on behalf of stakeholder, which is to reveal quality related information about the product with respect to the circumstances in which it is meant to operate. This includes and is not limited to the process of executing a program or application with the purpose of finding the errors. Quality is a value to some person and not absolute.

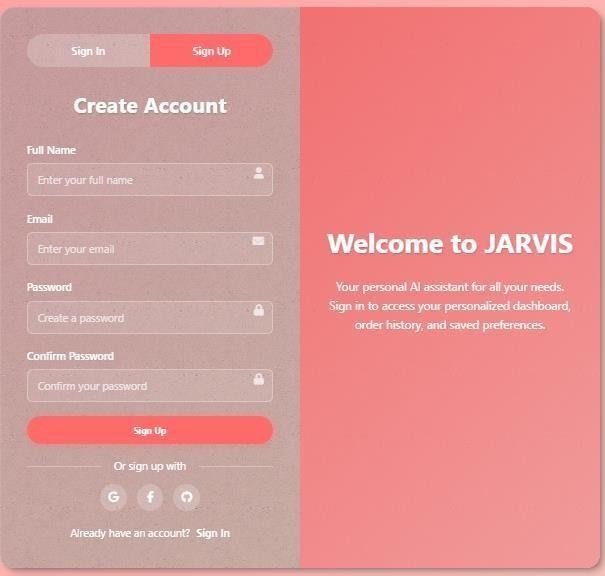
Testing can never completely begin the correctness of arbitrary computer software.[7] One of the e-commerce products that is popular nowadays is online shopping. Online shopping has been experiencing its golden years and becoming one of the potential contributions in e- commerce. The implication of this phenomenon is the spread of online shopping websites across the regional. One website originating from Egypt for example will be accessed and used by the people of all around the world. There will be a lot of advantages caused by this. Besides companies, customers gain their profits or advantages as well.

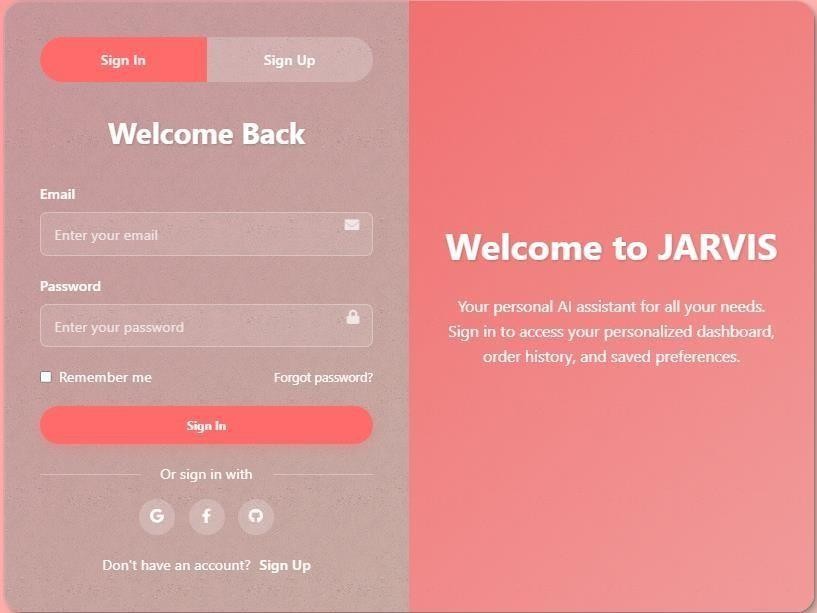
E-commerce has been considered as knowledge-based economy that can support the economy of the countries applying it [2]. The importance of e- commerce has caused high competitiveness among the online business doers. Many efforts have been done to discover the factors behind both success and failure of e-commerce specifically online shopping adoption. This includes the abundant studies which are related with online shopping, e- commerce, or cultural values in e-commerce and website design conducted by many researchers and practitioners. One of the important factors in the online shopping is the cultural factors.[8] Online Shopping Procedure:



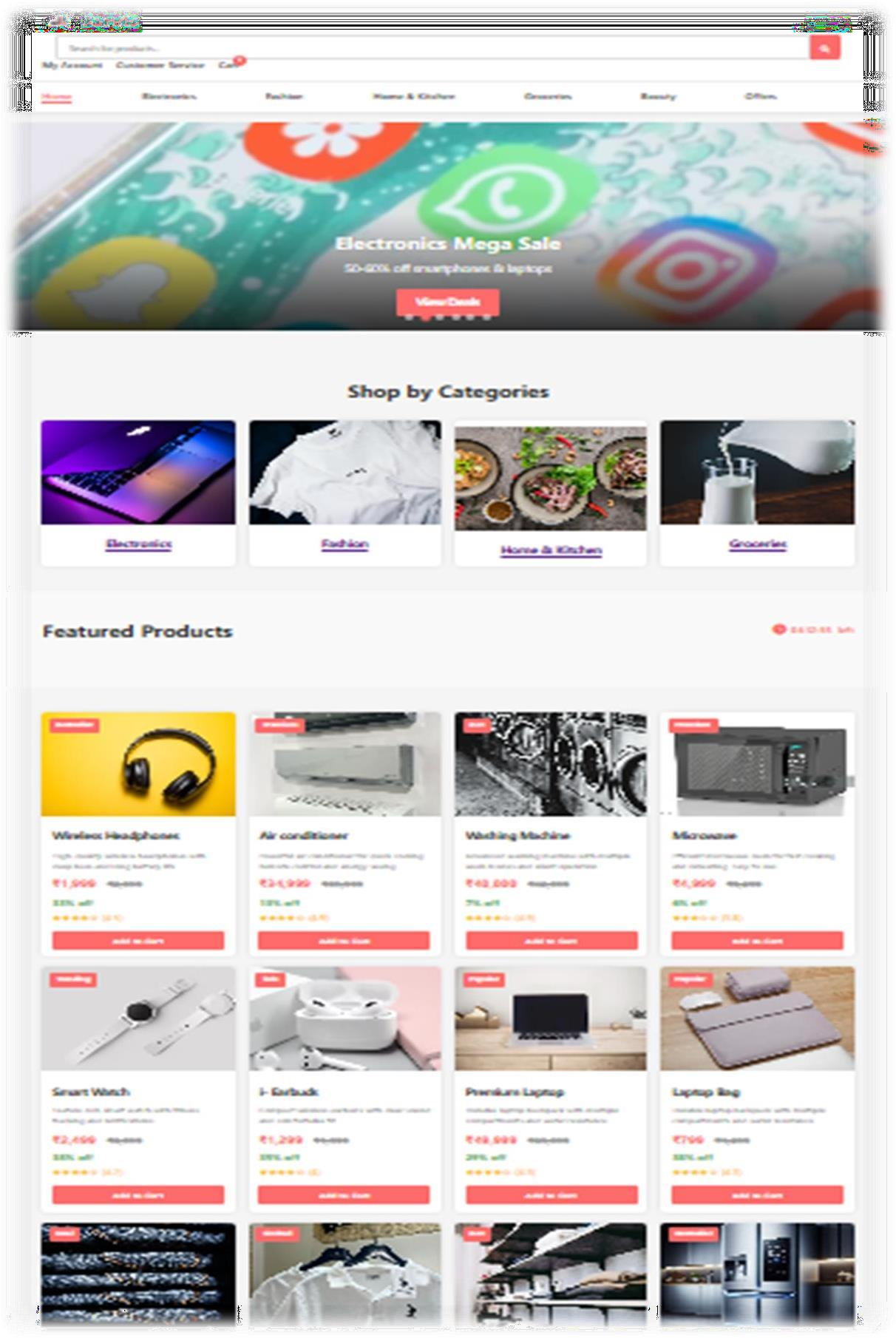
### SCREEN SHOTS

SIGN IN AND SIGN UP PAGE.

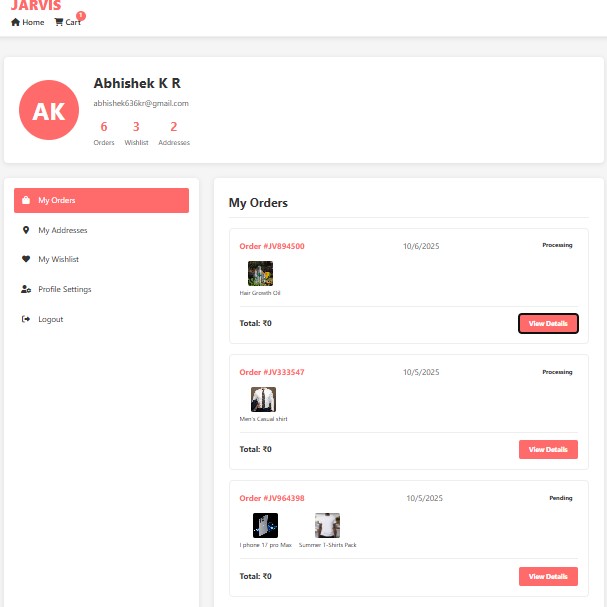




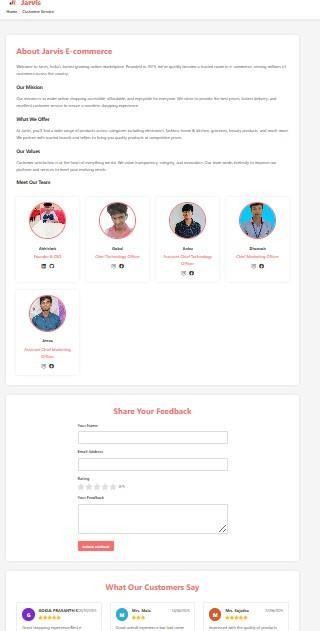
HOME PAGE.

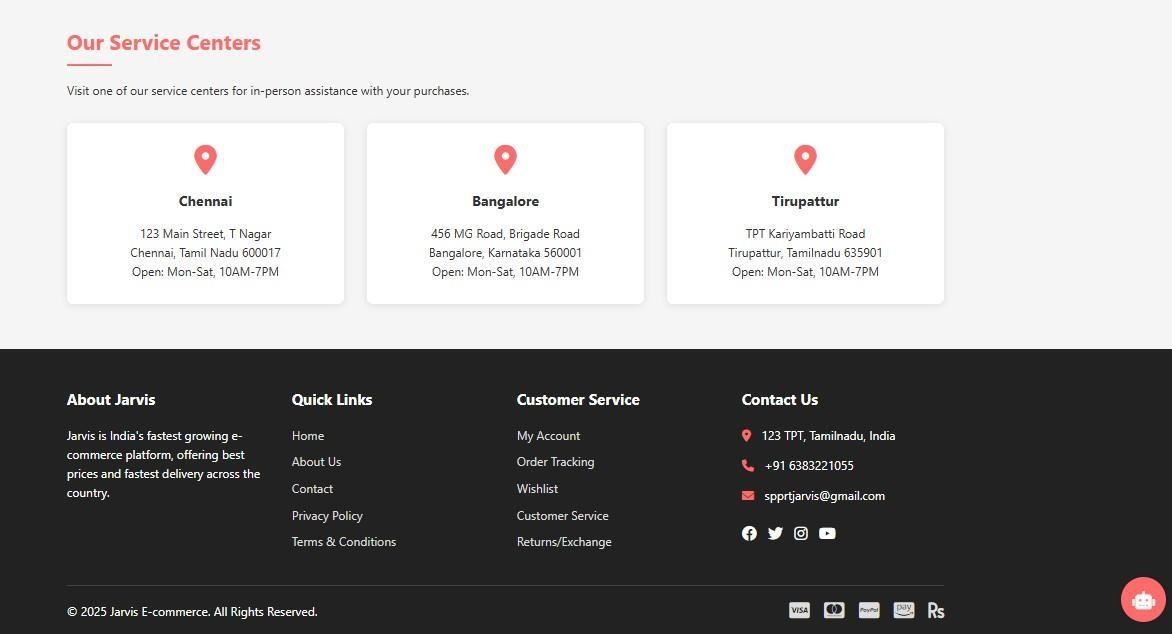
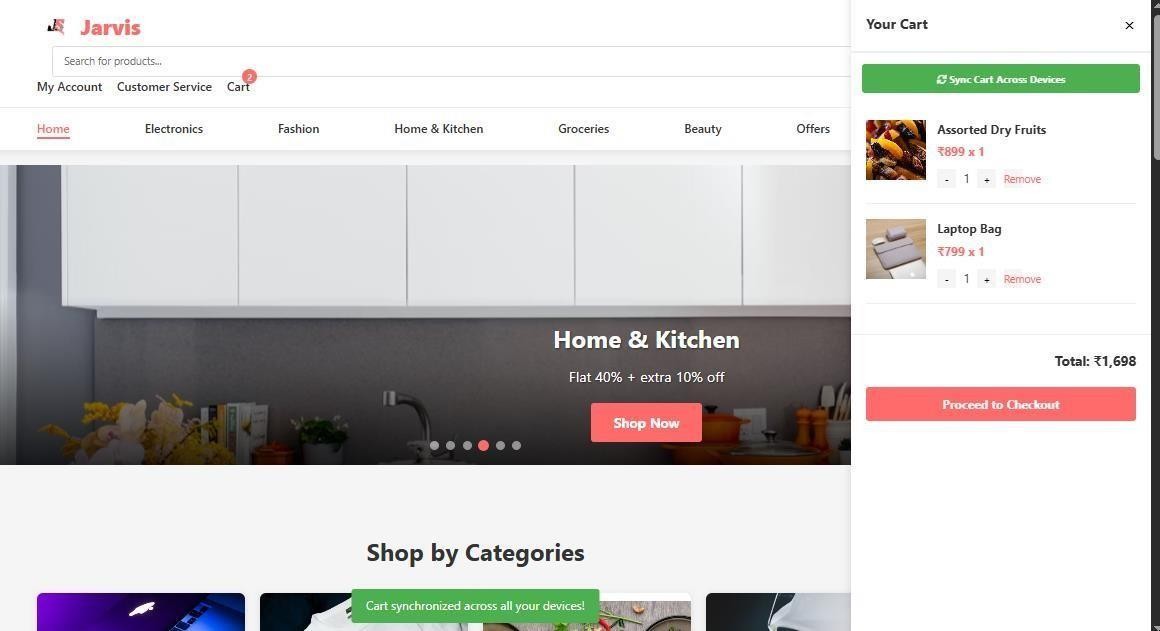


ACCOUNT PAGE.



ABOUT US PAGE.





CART

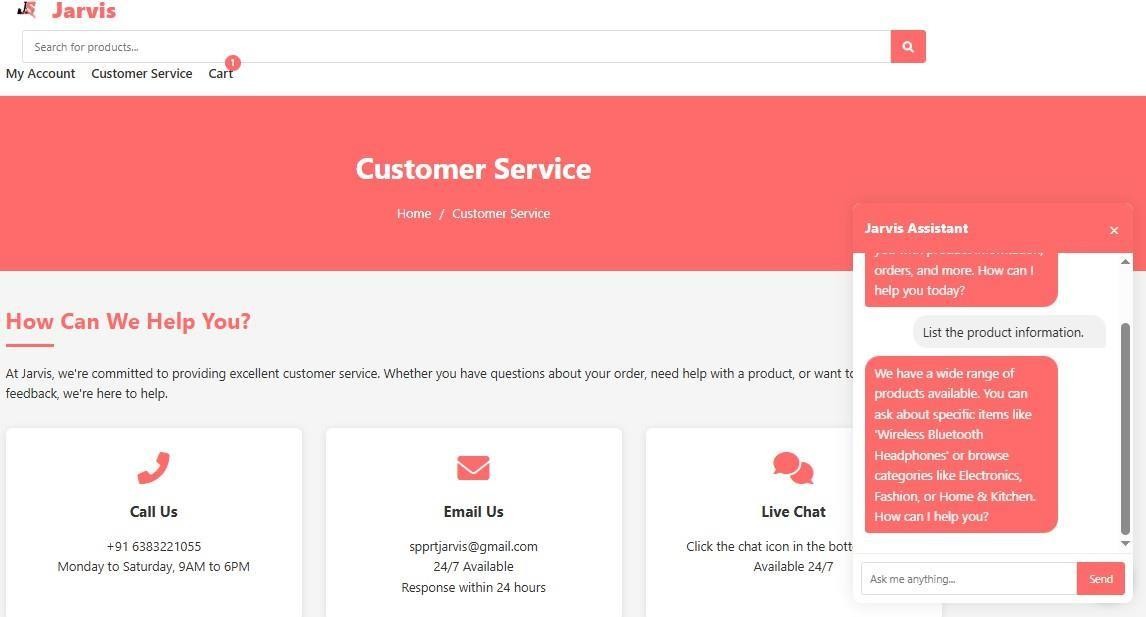
AND

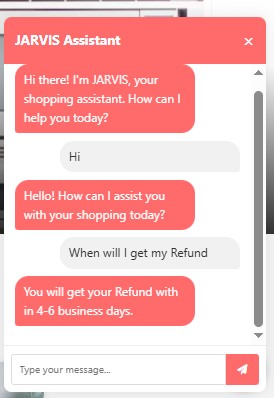
FOOTER

LINKS

PAGE.

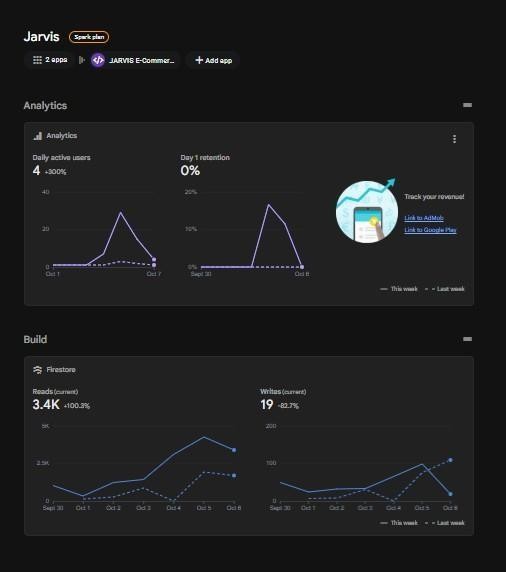
CUSTOMER SERVICE PAGE.

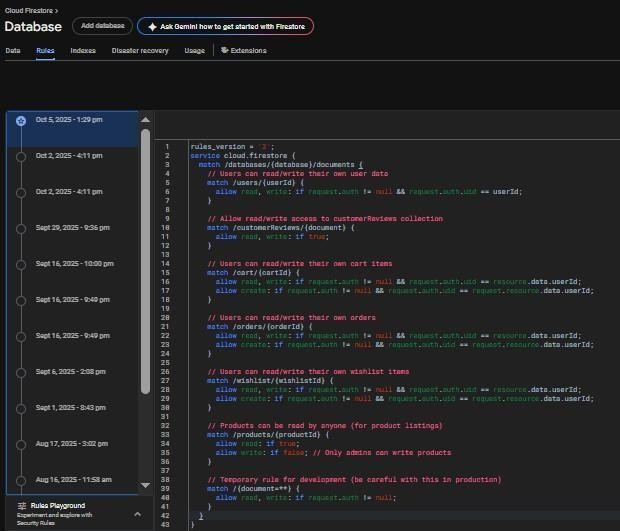




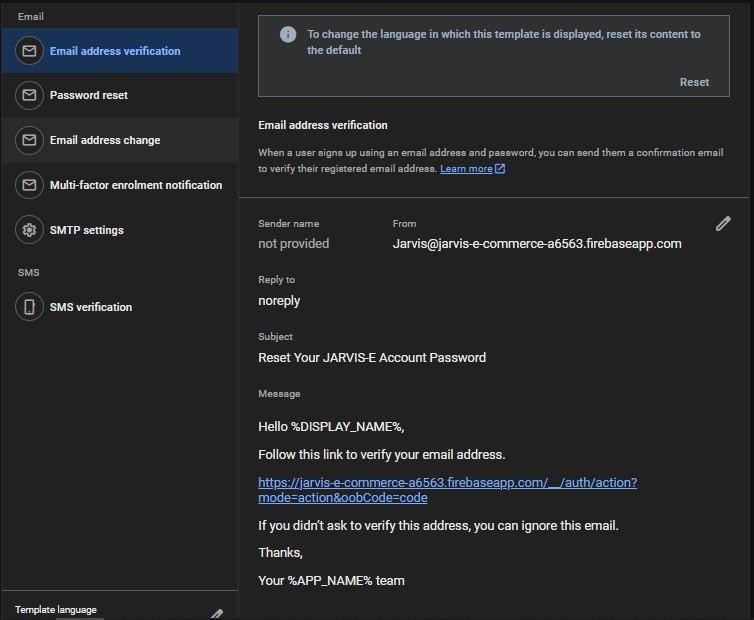
### DATABASE DESGIN

ANALYTICS.



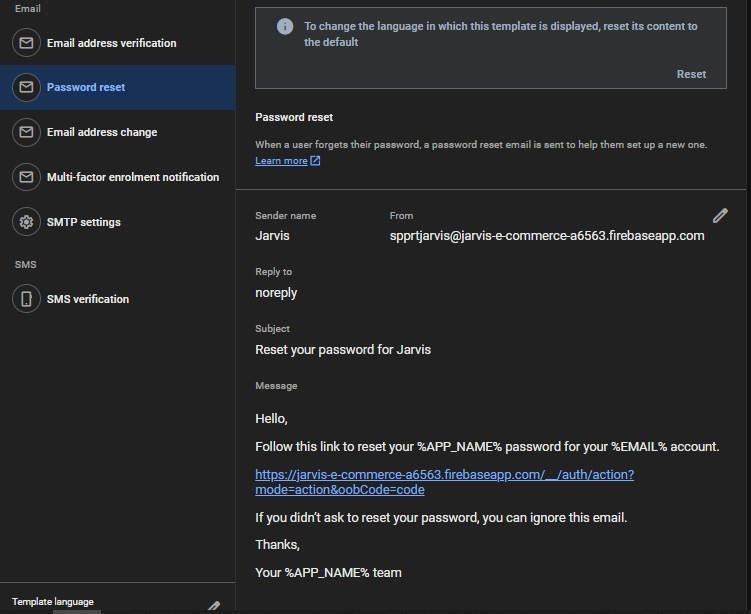


Database Email Verification.

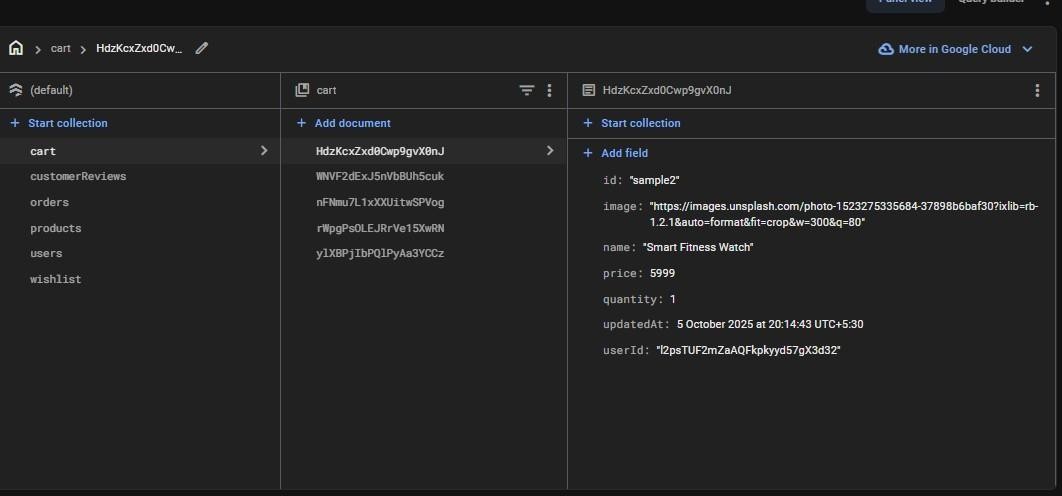


.

Password Reset Verification.



.



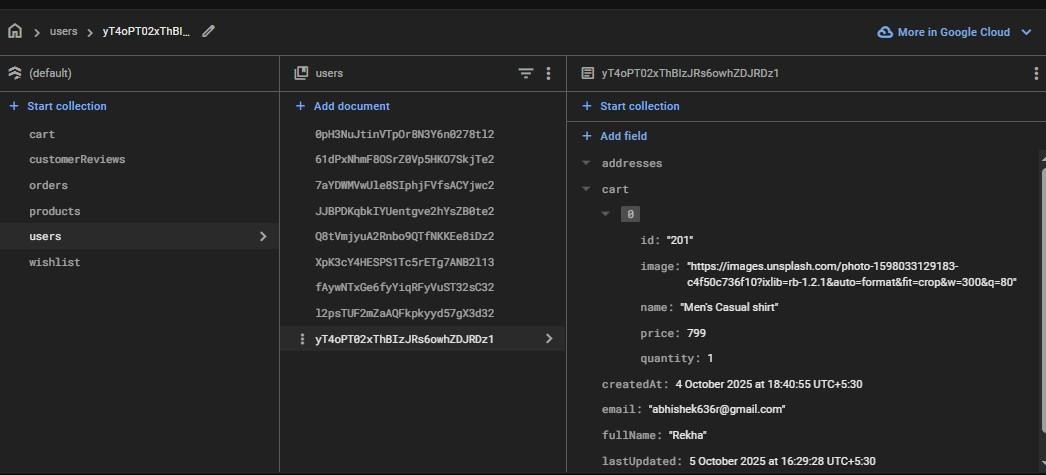
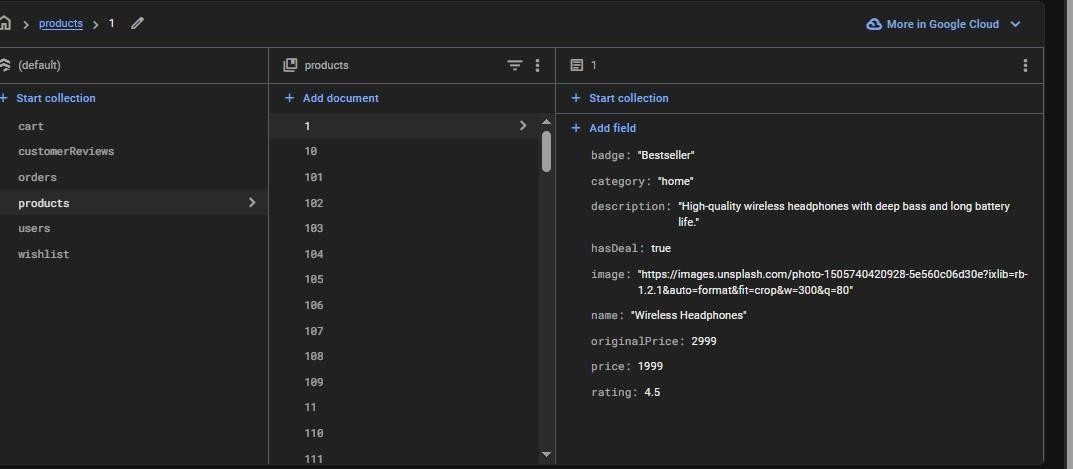
Database

Cart

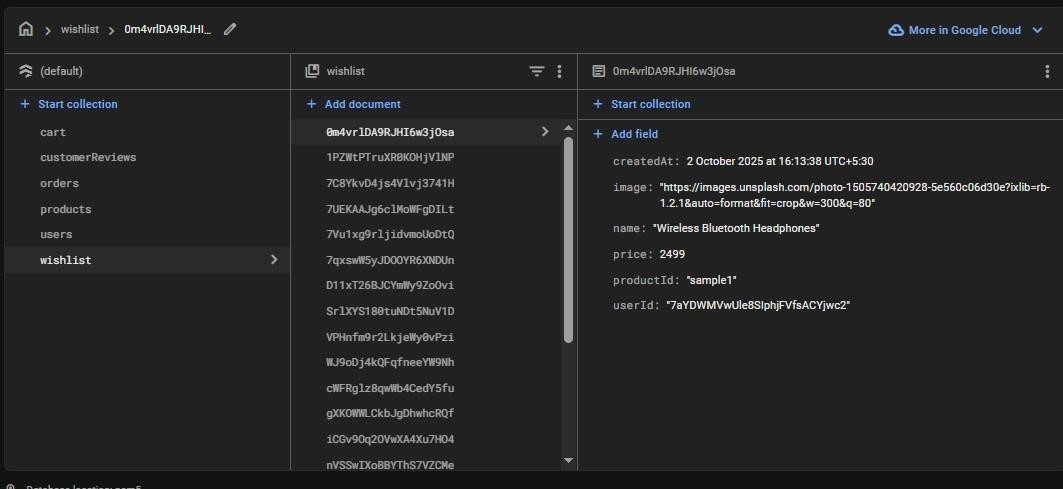
Collection.

.

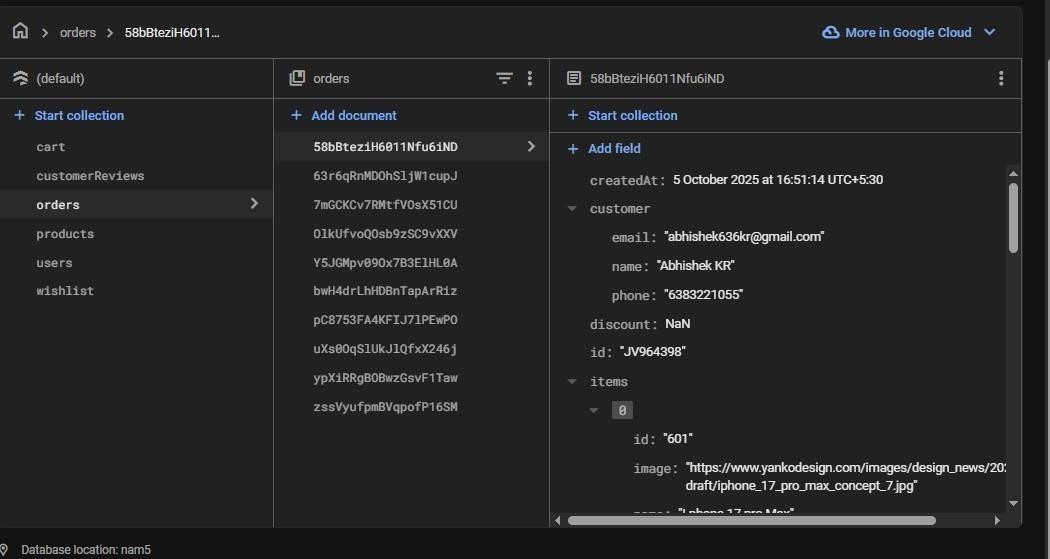
User & Products Details.



Database Orders History.



.



# CHAPTER V

**APPENDICES**

**FRONT-END CODE**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>JARVIS - Sign In / Sign Up</title>**

**<link rel="icon" type="image/png" href="top.jpg">**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0- beta3/css/all.min.css">**

**<script type="module" src="firebase-config.js" defer></script>**

**<style>**

**/\* Base Styles \*/**

**:root {**

**--primary-color: #ff6b6b;**

**--secondary-color: #333;**

**--light-color: #f5f5f5;**

**--dark-color: #222;**

**--text-color: #333;**

**--text-light: #666;**

**--white: #fff;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="auth-container glass3d">**

**<div class="auth-form">**

**<div class="auth-tabs">**

**<div class="auth-tab active" data-tab="signin">Sign In</div>**

**<div class="auth-tab" data-tab="signup">Sign Up</div>**

**</div>**

**<!-- Sign In Form -->**

**<div id="signin-form" class="form-container active">**

**<h2>Welcome Back</h2>**

**<form id="signinForm">**

**<**

**<div class="remember-me">**

**<input type="checkbox" id="remember-me">**

**<label for="remember-me">Remember me</label>**

**</div>**

**<a href="#" class="forgot-password">Forgot password?</a>**

**</div>**

**<button id="signin-submit" class="btn btn-submit">Sign In</button> <div class="social-login">**

**<p>Or sign in with</p>**

**<div class="social-icons">**

**<a href="#" class="social-icon" id="google-signin"><i class="fab fa-google"></i></a>**

**<a href="#" class="social-icon" id="facebook-signin"><i class="fab fa-facebook- f"></i></a>**

**<a href="#" class="social-icon" id="github-signin"><i class="fab fa-github" aria- hidden="true"></i></a>**

**</div>**

**</div>**

**<div class="auth-switch">**

**Don't have an account? <a href="#" class="switch-to-signup">Sign Up</a>**

**</div>**

**</form>**

**</div>**

**<!-- Sign Up Form -->**

**<div id="signup-form" class="form-container">**

**<h2>Create Account</h2>**

**<form id="signupForm">**

**<div id="signUpMessage" class="messageDiv" style="display: none;"></div>**

**<div class="form-group">**

**<label for="signup-name">Full Name</label>**

**<input type="text" id="signup-name" placeholder="Enter your full name">**

**<i class="fas fa-user input-icon"></i>**

**</div>**

**<div class="form-group">**

**<label for="signup-email">Email</label>**

**<input type="email" id="signup-email" placeholder="Enter your email">**

**<div class="social-icons">**

**<a href="#" class="social-icon" id="google-signup"><i class="fab fa-google"></i></a>**

**<h2>Welcome to JARVIS</h2>**

**<p>Your personal AI assistant for all your needs. Sign in to access your personalized dashboard, order history, and saved preferences.</p>**

**<!-- <a href="index.html" class="btn">Learn More</a> -->**

**</div>**

**</div>**

**</div>**

**<script>**

**// DOM elements const signInForm = document.getElementById('signin-form'); const signUpForm = document.getElementById('signup-form'); const signInTab = document.querySelector('[data- tab="signin"]'); const signUpTab = document.querySelector('[data-tab="signup"]'); const switchToSignUp = document.querySelector('.switch-to-signup');**

**const switchToSignIn = document.querySelector('.switch-to-signin'); const signIpMessage = document.getElementById('signIpMessage'); const signUpMessage = document.getElementById('signUpMessage');**

**// Switch between sign in and sign up signInTab.addEventListener('click', () => { signInTab.classList.add('active'); signUpTab.classList.remove('active'); signInForm.classList.add('active'); signUpForm.classList.remove('active');**

**function signInWithFacebook() {**

**// This will be implemented in firebase-config.js console.log('Sign in with Facebook'); showMessage(signIpMessage, "Facebook sign in will be implement"); element.style.opacity = '0';**

**}, 1000);**

**}, 5000);**

**}**

**</script>**

**</body>**

**</html>**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Jarvis - Indian E-commerce</title>**

**<link rel="icon" type="image/png" href="top.jpg">**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0- beta3/css/all.min.css">**

**<!-- Firebase SDK -->**

**<script src="https://**[**www.gstatic.com/firebasejs/9.6.0/firebase-app-compat.js"><**](http://www.gstatic.com/firebasejs/9.6.0/firebase-app-compat.js)**/script>**

**<script src="https://**[**www.gstatic.com/firebasejs/9.6.0/firebase-auth-compat.js"><**](http://www.gstatic.com/firebasejs/9.6.0/firebase-auth-compat.js)**/script>**

**<script src="https://**[**www.gstatic.com/firebasejs/9.6.0/firebase- firestore-**](http://www.gstatic.com/firebasejs/9.6.0/firebase-firestore-) **compat.js"></script>**

**<!-- Your Firebase config -->**

**<script src="firebase-config.js"></script>**

**<style>**

**/\* Base Styles \*/**

**:root {**

**--primary-color: #ff6b6b;**

**--secondary-color: #333;**

**--light-color: #f5f5f5;**

**--dark-color: #222;**

**--text-color: #333;**

**--text-light: #666;**

**--white: #fff;**

**}**

**\* { margin: 0; padding: 0; box-sizing: border-box; font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;**

**}**

**body { background-color: var(--light-color); color: var(--text-color); line-height: 1.6;**

**}**

**.container { width: 90%; max-width: 1200px; margin: 0 auto; padding: 0 15px;**

**}**

**.logo h1 { color: var(--primary- color); font-size: 28px; font-weight: 800;**

**}**

**.search-bar { display: flex; margin: 0 20px; flex-grow: 0.5; position: relative;**

**}**

**#search-results**

**{ display: none;**

**}**

**#search-results p { padding: 15px; color: #666;**

**}**

**#search-results div:hover { background-color:**

**#f5f5f5;**

**}**

**.search-bar input { width: 100%; padding: 10px 15px; border: 1px solid #ddd; border-radius: 4px 0 0 4px; font-size: 14px; outline: none; transition: border-color 0.3s;**

**}**

**.search-bar input:focus { border-color: var(--primary-color);**

**}**

**.search-bar button { padding: 10px 15px; background-color: var(--primary- color); color: var(--white); border: none; border-radius: 0 4px 4px 0; cursor: pointer; transition: background-color 0.3s;**

**}**

**.hero-content { text-align: center; color:**

**white; padding: 20px; background-color: rgba(0, 0, 0, 0.5); border-radius: 8px; max-width: 80%;**

**}**

**/\* Hide scrollbar for cleaner look \*/**

**.hero-carousel::-webkit- scrollbar { display: none;**

**}**

**/\* Hero Offer Gallery Styles \*/**

**.offer-gallery- container { position: relative; margin: 20px 0 40px; }**

**.offer-gallery { display: flex; overflow-x: auto; scroll-snap-type: x mandatory; scroll-behavior: smooth;**

**-webkit-overflow-scrolling: touch; gap: 0; padding: 0; background-color:**

**#f5f5f5; scrollbar-width: none;**

**}**

**.offer-gallery::-webkit- scrollbar { display: none;**

**}**

**.offer-card { scroll-snap-align: start; flex: 0 0 100%; position: relative; height: 400px; overflow: hidden;**

**}**

**.offer-card img { width: 100%; height: 100%; object-fit:**

**cover;**

**}**

**.offer-content { position: absolute; bottom: 0;**

**left: 0; right: 0; background: linear-gradient(to top, rgba(0, 0, 0, 0.8), transparent); color: white; padding: 30px; text-align: center;**

**}**

**.offer-content h3**

**{ text-align: center; margin-bottom: 10px; font-size: 2rem; margin- bottom: 30px; }**

**.featured-products h2 { font-size: 32px; color: #333; font-weight: 700;**

**}**

**.featured-products .time-left { display: flex; align-items: center; gap: 8px; color: #ff6b6b; font-weight:**

**500;**

**}**

**.featured-products .time-left i**

**{ font-size: 18px;**

**}**

**/\* Product Grid \*/**

**.featured-products .product-grid { display: grid; grid-template-columns: repeat(auto-fill, minmax(250px, 1fr)); gap: 20px;**

**}**

**.featured-products .product- card { background-color:**

**#fff;**

**border-radius: 8px; overflow:**

**hidden; box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1); position: relative; transition: transform 0.3s, box-shadow 0.3s;**

**}**

**.featured-products .product-card:hover { transform: translateY(-5px); box-shadow: 0 5px 15px rgba(0, 0, 0, 0.15);**

**}**

**.featured-products .product-card .badge { position: absolute; top:**

**10px; left: 10px; background-color: #ff6b6b; color: #fff; padding: 5px 10px; border- radius: 4px; font-size: 12px; font- weight: bold; z-index: 10;**

**}**

**.featured-products .product-card img**

**{ width: 100%; height:**

**200px; object-fit:**

**cover; transition: transform 0.3s;**

**}**

**.featured-products .product-card:hover img { transform: scale(1.05);**

**}**

**.featured-products .product-info { padding: 15px;**

**}**

**.featured-products .product-info h3 { margin-bottom: 10px;**

**font-size: 18px; color: #333; font-weight: 600; line-height: 1.3;**

**}**

**.featured-products .product-info**

**.price { font-weight: bold; color:**

**#ff6b6b; font-size:**

**20px; margin-bottom: 5px;**

**}**

**.featured-products .product-info .original- price { text-decoration: line-through; color: #999; font-size: 16px; margin-left: 10px;**

**}**

**.featured-products .product-info .discount { color: #4caf50; font-weight: bold; font-size: 14px;**

**margin-bottom: 10px;**

**}**

**.featured-products .product-info .rating**

**{ color: #ffc107; margin-bottom:**

**10px; font-size:**

**14px;**

**}**

**.featured-products .product-info .rating span { color: #666; margin-left: 5px;**

**}**

**.featured-products .product-info .add-to-cart { width: 100%; padding: 10px; background-color: #ff6b6b; color: #fff; border: none; border-radius:**

**4px; cursor:**

**pointer; font- weight: bold; transition: background-color 0.3s; }**

**.featured-products .product-info .add-to- cart:hover { background-color: #ff5252;**

**}**

**/\* Responsive Styles \*/ @media (max-width:**

**992px) {**

**.featured-products .product-grid { grid-template-columns: repeat(auto-fill, minmax(220px, 1fr));**

**}**

**}**

**@media (max-width: 768px) {**

**.featured-products .section- header { flex-direction: column; align-items: flex-start; gap: 15px;**

**}**

**.featured-products .product-grid { grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));**

**}**

**}**

**@media (max-width: 576px) {**

**.featured-products .product- grid { grid-template-columns: 1fr;**

**}**

**.featured-products h2 { font-size: 28px;**

**}**

**}**

**.no-products,**

**.empty-cart { text-align: center; padding: 40px; color: #666; font-size: 18px; grid-column: 1 /**

**-1;**

**}**

**product-card img { width:**

**100%; height: 200px; object- fit: cover; transition: transform 0.3s;**

**}**

**font-size: 18px; grid-column: 1 /**

**-1;**

**}**

**/\* Responsive User Actions \*/ @media (max-width: 768px) {**

**.user-actions .desktop-text { display: none;**

**}**

**.user-actions .mobile-icon { display: inline-block;**

**}**

**.user-actions a { padding:**

**}**

**/\* .menu-toggle { display: block;**

**} \*/**

**.menu-toggle { display: block !important; visibility: visible !important; opacity: 1 !important;**

**}**

**nav ul { display: none;**

**}**

**}**

**@media (max-width: 768px) { nav .container { display: block !important; overflow-x: auto !important; white-space: nowrap**

**!important;**

**-webkit-overflow-scrolling: touch !important;**

**}**

**nav ul { display: flex !important; flex-wrap: nowrap !important; overflow-x: auto !important; padding-bottom: 10px !important;**

**}**

**nav ul li { margin-right: 30px**

**!important; flex-shrink: 0**

**!important;**

**}**

**\*/**

**/\* Hide mobile menu elements since we don't need them \*/**

**/\* .menu-toggle, .mobile-menu { display: none !important;**

**} \*/**

**.menu-toggle, .mobile-menu { display: block !important;**

**}**

**}**

**</style>**

**</head>**

**<body>**

**<header>**

**<div class="top-bar">**

**<div class="container">**

**<div class="logo">**

**<h1><img src="logo.jpg" style="width: 50px; height: 50px; border-radius: 50%; object- fit: cover;" alt="Jarvis Logo"> Jarvis</h1>**

**</div>**

**<div class="search-bar">**

**<input type="text" placeholder="Search for products...">**

**<button><i class="fas fa-search"></i></button>**

**</div>**

**<div class="user-actions">**

**<a href="profile.html">**

**<i class="fas fa-user mobile-icon"></i>**

**<span class="desktop-text">My Account</span>**

**</a>**

**<a href="Customer.html">**

**<i class="fas fa-headset mobile-icon"></i>**

**<span class="desktop-text">Customer Service</span>**

**</a>**

**<a href="#" class="cart">**

**<i class="fas fa-shopping-cart mobile-icon"></i>**

**<span class="desktop-text">Cart</span>**

**<span id="cart-count">0</span>**

**</a>**

**</div>**

**<div class="chatbot-container" id="chatbot-container">**

**<div class="chatbot-header">**

**<h3>JARVIS Assistant</h3>**

**<button class="close-chatbot" id="close-chatbot">&times;</button>**

**</div>**

**<div class="chatbot-messages" id="chatbot-messages">**

**<div class="message bot-message">**

**Hi there! I'm JARVIS, your shopping assistant. How can I help you today?**

**</div>**

**</div>**

**<div class="chatbot-input">**

**<input type="text" id="chatbot-input" placeholder="Type your message...">**

**<button id="send-message"><i class="fas fa-paper-plane"></i></button>**

**</div>**

**</div>**

**<script>**

**// Firebase initialization const firebaseConfig = { apiKey: "AIzaSyDqXN\_NjP3OUGhd-ah- hWTO66REtlMqwAk", authDomain: "jarvis-e- commercea6563.firebaseapp.com", projectId: "jarvis-e- commerce-a6563", storageBucket: "jarvis-e-commerce- a6563.firebasestorage.app", messagingSenderId:**

**"998302088355", appId: "1:998302088355:web:36279bf2694647baf05316", measurementId: "G-Q9E7Z27L6K"**

**};**

**// Initialize Firebase firebase.initializeApp(firebaseConfig); const db = firebase.firestore();**

**// DOM Elements const menuToggle = document.getElementById('menu- toggle'); const mobileMenu = document.getElementById('mobile-menu'); const cartSidebar**

**= document.getElementById('cart-sidebar'); const closeCart = document.getElementById('close-**

**cart');**

**const cartCount = document.getElementById('cart- count'); const cartItems = document.getElementById('cart-items'); const cartTotal =**

**document.getElementById('cart-total');**

**const chatbotToggle = document.getElementById('chatbot-toggle');**

**const closeChatbot = document.getElementById('close-chatbot'); const sendMessage = document.getElementById('send- message'); const chatbotInput = document.getElementById('chatbot-input'); const chatbotMessages = document.getElementById('chatbot- messages'); const featuredProductsGrid = document.getElementById('featured-products'); const dealProductsGrid = document.getElementById('deal-products'); const syncCartBtn = document.getElementById('sync-**

**cart');**

**// Cart array let cart = JSON.parse(localStorage.getItem('cart')) || [];**

**// Mobile Menu Toggle menuToggle.addEventListener('click', () => { mobileMenu.classList.toggle('active');**

**});**

**// Cart Toggle document.querySelector('.cart').addEventListener('click', (e) => { e.preventDefault(); cartSidebar.classList.add('active');**

**});**

**closeCart.addEventListener('click', () => {**

**cartSidebar.classList.remove('active');**

**});**

**// Chatbot Toggle chatbotToggle.addEventListener('click', () => { chatbotContainer.classList.toggle('active');**

**});**

**closeChatbot.addEventListener('click', () => { chatbotContainer.classList.remove('active');**

**});**

**// Send Message sendMessage.addEventListener('click', ()**

**=> { const message = chatbotInput.value.trim(); if (message)**

**{**

**addMessage(message, 'user'); chatbotInput.value = ''; respondToMessage(message);**

**}**

**});**

**chatbotInput.addEventListener('keypress', (e)**

**=> { if (e.key === 'Enter') {**

**sendMessage.click();**

**}**

**});**

**// Add Message to Chat function addMessage(text, type) { const messageDiv = document.createElement('div'); messageDiv.classList.add('message', `${type}message`); messageDiv.textContent = text; chatbotMessages.appendChild(messageDiv); chatbotMessages.scrollTop = chatbotMessages.scrollHeight;**

**}**

**// Chatbot Responses function respondToMessage(message) { let response = ''; if (message.toLowerCase().includes('hello') || message.toLowerCase().includes('hi')) { response =**

**'Hello! How can I assist you with your shopping today?';**

**} if (message.toLowerCase().includes('product') || message.toLowerCase().includes('item')) { response = 'I can help you find products. What are you looking for?';**

**} if (message.toLowerCase().includes('electronics')) {**

**response = 'We have a wide range of electronics including smartphones, laptops, and accessories. Check out our electronics section!';**

**} if (message.toLowerCase().includes('refund')) {**

**response = 'You will get your Refund with in 4-6 business days.';**

**} if (message.toLowerCase().includes('groceries')) {**

**response = 'Our groceries items delivery with one day and more info check out our "groceries" section.';**

**} if (message.toLowerCase().includes('fashion')) {**

**response = 'Our fashion collection includes clothing, footwear, and accessories for men, women, and kids.';**

**} if (message.toLowerCase().includes('delivery') || message.toLowerCase().includes('shipping')) { response = 'We offer free shipping on orders above ₹499. Delivery usually takes 3-5 business days.'; } if (message.toLowerCase().includes('return') || message.toLowerCase().includes('exchange')) { response = 'We have a 7-day return policy for most items. Some conditions apply.';**

**} if (message.toLowerCase().includes('discount') || message.toLowerCase().includes('offer')) { response = 'Check out our "Offers" section for current discounts and promotions.';**

**} if (message.toLowerCase().includes('thank you') || message.toLowerCase().includes('thanks')) {**

**response = 'I\'m here to help with your shopping needs. You can ask me about products, delivery, returns, or current offers.';**

**}**

**if**

**(!response)**

**{ response**

**= [**

**"I'm not sure about that!. Could you rephrase that?",**

**"Can I connect the chat with Human!. Could you ask me about our products or return?",**

**"I'm a shopping assistant. Ask me about products, orders, or shipping!",**

**"I don't have that information right now, but you can contact our customer service team for help."**

**<!-- Firebase SDKs -->**

**<script src="https://**[**www.gstatic.com/firebasejs/9.23.0/firebase-app.js"><**](http://www.gstatic.com/firebasejs/9.23.0/firebase-app.js)**/script>**

**<script src="https://**[**www.gstatic.com/firebasejs/9.23.0/firebase-firestore.js"><**](http://www.gstatic.com/firebasejs/9.23.0/firebase-firestore.js)**/script>**

**<!-- Your Firebase config -->**

**<script type="module" src="./firebase-config.js"></script>**

**<!-- Add this import after your firebase-config import -->**

**<script type="module" src="cart-functions.js"></script>**

**</script>**

**</body>**

**</html>**

**DATABASE.**

// Your web app's Firebase configuration const firebaseConfig =

{

apiKey: "AIzaSyDqXN\_NjP3OUGhd-ah- hWTO66REtlMqwAk", authDomain: "jarvis-e- commercea6563.firebaseapp.com", projectId:

"jarvis-e-commerce-a6563", storageBucket: "jarvis-e-commerce- a6563.firebasestorage.app", messagingSenderId: "998302088355", appId: "1:998302088355:web:36279bf2694647baf05316", measurementId: "G-Q9E7Z27L6K"

};

export { firebaseConfig };

// Import the functions you need from the SDKs you need import { initializeApp } from "https://[www.gstatic.com/firebasejs/12.1.0/firebase- app.js"](http://www.gstatic.com/firebasejs/12.1.0/firebase-app.js); import { getAnalytics } from "https://[www.gstatic.com/firebasejs/12.1.0/firebase-analytics.js"](http://www.gstatic.com/firebasejs/12.1.0/firebase-analytics.js); import { getAuth, createUserWithEmailAndPassword, signInWithEmailAndPassword,

GoogleAuthProvider, FacebookAuthProvider, GithubAuthProvider, signInWithPopup, sendPasswordResetEmail, onAuthStateChanged, signOut, updateProfile

} from "https://[www.gstatic.com/firebasejs/12.1.0/firebase-auth.js"](http://www.gstatic.com/firebasejs/12.1.0/firebase-auth.js); import { getFirestore, collection, addDoc, doc, setDoc, getDoc, query, where, getDocs, updateDoc, arrayUnion, arrayRemove, deleteDoc

} from "https://[www.gstatic.com/firebasejs/12.1.0/firebase-firestore.js"](http://www.gstatic.com/firebasejs/12.1.0/firebase-firestore.js);

// Initialize Firebase

const app = initializeApp(firebaseConfig); const analytics = getAnalytics(app); const auth = getAuth(app); const db = getFirestore(app); const googleProvider = new GoogleAuthProvider(); const facebookProvider = new

FacebookAuthProvider(); const githubProvider = new GithubAuthProvider();

// Make Firebase functions available globally window.firebaseAuth = auth; window.firebaseDb = db; window.firebaseFunctions = { getDoc, setDoc, updateDoc, doc, collection, addDoc, query, where, getDocs, arrayUnion, arrayRemove, deleteDoc, signOut, updateProfile, onAuthStateChanged

};

// Function to show messages function showMessage(message, elementId) { const messageDiv = document.getElementById(elementId); if (!messageDiv) return;

messageDiv.textContent = message;

messageDiv.style.display = 'block'; messageDiv.style.opacity = '1'; setTimeout(() => { messageDiv.style.opacity = '0'; setTimeout(() => { messageDiv.style.display = 'none';

}, 500);

}, 5000);

}

// Function to update user profile async function updateUserProfile(userId, userData) { try { await updateDoc(doc(db, "users", userId), userData); return true;

} catch (error) { console.error("Error updating user profile:", error); return false;

}

}

// Function to save address to user's address book async function saveUserAddress(userId, addressData) { try {

const addressId = Date.now().toString(); const addressWithId = { ...addressData, id: addressId };

// Add address to user's address book await updateDoc(doc(db, "users", userId), { addresses:

arrayUnion(addressWithId)

});

// If this is set as default, update all other addresses to not be default if (addressData.isDefault) { const userDoc = await getDoc(doc(db, "users", userId)); if (userDoc.exists()) { const userData = userDoc.data();

const addresses = userData.addresses || [];

// Update all other addresses to not be default const updatedAddresses = addresses.map(addr => { if (addr.id !== addressId) { return { ...addr, isDefault: false };

}

return addr;

});

// Update the addresses array

await updateDoc(doc(db, "users", userId), { addresses: updatedAddresses

});

}

}

return true; } catch (error) { console.error("Error saving address:", error); return false;

}

}

// Function to delete user address async function deleteUserAddress(userId, addressId) { try { const userDoc = await getDoc(doc(db, "users", userId)); if (userDoc.exists()) { const userData = userDoc.data();

const addresses = userData.addresses || [];

// Filter out the address to delete const updatedAddresses = addresses.filter(addr => addr.id !== addressId); await updateDoc(doc(db, "users", userId), { addresses: updatedAddresses

});

return true;

}

return false; } catch (error) { console.error("Error deleting address:", error); return false;

}

}

// Function to get user's order history async function getUserOrderHistory(userId) { try {

// Query orders for this user const q = query(collection(db, "orders"), where("userId", "==", userId)); const querySnapshot = await getDocs(q);

const orders = []; querySnapshot.forEach((doc) => { orders.push({

id: doc.id,

...doc.data()

});

});

// Sort by date (newest first) orders.sort((a, b) => { const dateA = a.createdAt?.toDate?.() || new Date(0); const dateB = b.createdAt?.toDate?.() || new Date(0); return dateB - dateA;

});

return orders; } catch (error) { console.error("Error getting order history:", error); return [];

}

}

// Function to get user's saved addresses async function getUserAddresses(userId) { try { const userDoc = await getDoc(doc(db, "users", userId)); if (userDoc.exists()) { return userDoc.data().addresses || [];

}

return []; } catch (error) { console.error("Error getting user addresses:", error); return [];

}

}

// Function to set default address async function setDefaultAddress(userId, addressId)

{ try {

const userDoc = await getDoc(doc(db, "users", userId)); if (userDoc.exists()) { const userData = userDoc.data();

const addresses = userData.addresses || [];

// Update all addresses - set the selected one as default, others as not default const updatedAddresses = addresses.map(addr => { if (addr.id === addressId) { return { ...addr, isDefault: true };

} else { return { ...addr, isDefault: false };

}

});

// Update the addresses array await updateDoc(doc(db, "users", userId), { addresses: updatedAddresses

});

return true;

}

return false; } catch (error) { console.error("Error setting default address:", error); return false;

}

}

// Export functions for use in other files window.profileFunctions = { updateUserProfile, saveUserAddress, deleteUserAddress, getUserOrderHistory, getUserAddresses, setDefaultAddress

};

## POINTS NEED TO COVER IN E-SHOPPING WEBSITE TESTING

--For testing online shopping website the following important functions or pages to be tested: **Main pages, Product category pages, Product detail pages, Product search, Shopping basket, Checkout and Payment Systems.**

--In other important conditions need to be tested are;

**Browser Compatibility, Mobile Device Compatibility, Performance,**

**1 Main Pages Testing Components**

* Home page
* Featured products
* Special offers
* Information pages- About page, Shipping information, Returns policy, Terms page, Privacy policy

**2 Product Category Pages Testing Components**

* Any filters such as product filters, colors, sizes, types of product, etc.
* Any ability to sort products by name, price, size, etc.
* Add to shortlist or wish list facility.
* Add to basket.

**3 Product Detail Pages Testing Components**

* Product title
* Product description
* Product images
* Enlarge image
* Related products
* Any further product information, colors, sizes, options, extras.
* Add to shopping basket

**4 Product Search Testing Component**

* Keyword search- It may contains different options to select search categories

**5 Shopping Basket Testing Components**

* Add products to basket
* Remove product from basket
* Change quantities
* Select delivery option
* Check VAT and delivery costs add up correctly

1. **Checkout and Payment Systems**

As this is a testing plan for an ecommerce site then a specific section of the plan should be devoted to the checkout and payment area of the website. Test the checkout process including the following aspects: Final amount to pay – make sure that this value is correct, after the price of the products, VAT, delivery and any other charges. Test making changes to the products being ordered, changing delivery options, etc. and make sure that this final amount updates correctly

**Next is payments:**

Carry out a test payment using each payment method that you are offering such as debit cards, credit cards, Google Checkout, etc. Your payment system will most likely still be in test mode before you launch so all test payments will be carried out using the test payment details. An example list of payment types is as follows:

* + Place UPI payment
  + Place Rupay Debit card payment

1. **Browser Compatibility**

Extremely important, especially for ecommerce sites, as if the site doesn’t work in a particular browser then nobody using that browser can buy anything from you. This list of browsers will change, as new browsers are released and older browsers aren’t used as much (here’s hoping that IE7 goes away soon). Currently, we are testing all of the functionality listed above in each of the following web browsers: Internet Explorer 7 Internet Explorer 8 Internet Explorer 9 Mozilla Firefox (latest version) Google Chrome (latest version) Safari

1. **Mobile Device Compatibility**

If you wish your website to be used by visitors viewing it on their smartphone or tablet then you also need to test all of the functionality on each of the following mobile devices:

* Apple iPhone – 5, 5S/5C, 6 and 6 Plus
* Apple iPad – iPad 2, 3, 4, iPad Air and iPad Air 2)
* Apple iPad Mini – 1, 2 and 3
* Android Smartphone – such as Samsung Galaxy S5
* Android Tablet – such as Google Nexus 7
* Windows Phone – such as Nokia Lumia 635

**9 Performance Testing**

Your ecommerce website needs to load quickly in order for potential customers to not get frustrated and head elsewhere. Carry out a performance test using an online tool such as Webpage test and pay attention to what it tells you. For doing performance testing majority we can use Win runner and Load runner tools.

**Why E-commerce Application Testing is Important?**

The first and primary reason is because e-commerce is, by its very nature, business critical and highly visible to its user’s. Any failure can be immediately expensive in terms of lost revenue and even more expensive in the longer term if disaffected users seek alternative sites. E-commerce is a massive and growing market place but one which requires large up-front investment to enter successfully. The history of e-commerce development has been littered with expensive failures, at least some of which could have been avoided by better testing before the site was opened to the general public. Quality Assurance of the software or application developed Verification and validating the product/application before it goes live in the market to prevent it from intruders and hackers. Defect free and user friendly application

## Future Scope

* Our designed online shopping system provides a 24×7 service that is customers cansurf the website, place orders anytime they wish to. Also, the delivery system works 24×7 hours a week.
* If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as flipkart or eBay. Since the application is available in the Smartphone it is easily accessible and always available.
* Some of the features that can be modified and added to this system in the future involve its implementation by local shopkeepers, where shops will be providing an onlineinterface to customers for shopping and placing orders. Then some delivery persons can perform their work. This will be adding on benefit for the customers as it will save their time, plus it adds on for the shopkeepers .
* In future our job management system will have extra facilities and more companies for more employees.
* It will grant more easy access.
* We are going to include recommendation system to this project so it will be helpful for the user to search and apply for jobs as they wish.

## Conclusion

Technology has made significant progress over the years to provide consumers a better online shopping experience and will continue to do so for years to come. With the rapid growth of products and brands, people have speculated that online shopping will overtake in-store shopping. While this has been the case in some areas, there is still demand for Products stores in market areas where the consumer feels more comfortable seeing and touching the product being bought.

However, the availability of online shopping has produced a more educated consumer that can shop around with relative ease without having to spend a large amount of time. In exchange, online shopping has opened up doors to many small retailers that would never be in business if they had to incur the high cost of owning a Products store. At the end, it has been a win-win situation for both consumer and sellers

E-Commerce is not an IT issue but a whole business undertaking. Companies that use it as a reason for completely re-designing their business processes are likely to reap the greatest benefits. Moreover, E-Commerce is a helpful technology that gives the consumer access to business and companies all over the world