AMAZON CLONE

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by
Ashu Tyagi
2200290140041
Ashutosh Khagwal
2200290140042

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Mr. Praveen Kumar Gupta
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS
Uttar Pradesh-201206
(2023-2024)

CERTIFICATE

Certified that **Ashu Tyagi 2200290140041, Ashutosh Khagwal 2200290140042** has/ have carried out the project work having "**AMAZON CLONE**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr A.P.J ABDUL KALAM TECHNICAL UNIVERSITY (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Ashu Tyagi
Roll No 2200290140041
Ashutosh Khagwal
Roll No 2200290140042

Mr. Praveen Kumar Gupta
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr Arun Kumar Tripathi
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The Amazon Clone Project aims to replicate the core functionality and user experience of the renowned e-commerce platform, Amazon, by developing a scalable and robust e-commerce application. This project leverages modern web development technologies to create a user-friendly interface, seamless navigation, and a comprehensive set of features that closely emulate the original Amazon platform.

The project's core features encompass a comprehensive user authentication and authorization system, ensuring secure access and safeguarding user data. A dynamic product catalogue lies at the heart of the application, offering users an extensive array of products completed with detailed descriptions, images, and pricing information. The shopping experience is further enriched by a responsive shopping cart system and a streamlined checkout process, supporting various payment options for user convenience.

In line with Amazon's commitment to transparency and customer feedback, the clone project integrates a user reviews and ratings system. This feature empowers users to share their experiences, enhancing the credibility of the platform. The implementation of a robust search engine, complemented by intuitive filters, enables users to swiftly locate products based on specific criteria such as categories and brands.

In summary, the Amazon Clone Project aspires to deliver a comprehensive, secure, and scalable e-commerce solution, providing a solid foundation for further customization and expansion. Through a meticulous integration of modern technologies and design principles, the project aims to emulate the seamless and engaging online shopping experience that defines the original Amazon platform.

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Mr. Praveen Kumar Gupta** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr Arun Kumar Tripathi, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Date:

Ashu Tyagi
Ashutosh Khagwal

DECLARATION

I hereby declare that the work presented in this report entitled "**Amazon Clone**", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Date:

Ashu Tyagi
Ashutosh Khagwal

TABLE OF CONTENTS

Contents	Page No.
Certificate	i
Abstract	ii
Acknowledgement	iii
Declaration	iv
List of Figures	viii
CHAPTER 1: INTRODUCTION	**1-4**
1.1 Project Objective	1
1.2 Project Overview	1
1.3 Project Scope	2
1.4 Study of the System	2
1.4.1 Modules	2
1.4.1.1 Users	3
CHAPTER 2: REQUIREMENTS AND SYSTEM ANALYSIS	**5-11**
2.1 Existing System	5
2.2 Proposed System	6
2.3 System Requirement Specification	6
2.3.1 General Description	6
2.3.2 System Objectives	6
2.3.3 System Requirements	7
2.3.3.1 Non-Functional Requirements	7
2.3.3.2 Functional Requirements	7

2.3.3.3 Hardware and Software Requirements 8
2.4 Feasibility Analysis 9
2.5 Planning and Scheduling 9
2.6 Context Design 11
CHAPTER 3: DESIGN OF THE SYSTEM 12-23
3.1 Input and Output Design 12
3.1.1 Input Design 12
3.1.2 Output Design 13
3.2 Database 13
3.3 System Tools 13
3.3.1 Frontend 13
3.3.2 Backend 14
3.5 ER Diagrams 16
3.6 Data Flow Diagram 21
CHAPTER 4: METHODOLOGY 24-30
4.1 Project Stages 24
4.2 High-Level Approach 24
4.3 Primary Goals 25
4.4 Environment Setup 25
4.5 Navigation with the React Router 27
4.6 Creating the navigation bar 27
4.7 Creating the homepage 28
4.8 Setting Up React Context API 29

4.9 Adding the Basket Functionality 29

4.10 Setting Up Firebase and Authentication 30

4.11 The Checkout 30

4.12 Payments Functionality and more 30

CHAPTER 5: IMPLEMENTATION AND TESTING 31-33

5.1 Implementation Approaches 31

5.2 Code Efficiency 31

5.3 Testing Approach 31

5.3.1 White Box Testing 33

5.3.2 Black Box Testing 33

CHAPTER 6: RESULT 34-37

6.1 Screenshots 34

6.1.1 Homepage 34

6.1.2 Sign in Page 35

6.1.3 Shopping Cart 36

6.1.4 Payment Page 37

CHAPTER 7: CONCLUSION 38-39

7.1 Conclusion 38

7.2 Future Enhancement 39

REFERENCES 40

LIST OF FIGURES

Figure No.	Name of Figure	Page No.
1.1	User Module	3
2.1	Gantt Chart for Project	10
2.6	Context Design	11
3.1	Login	16
3.2	User Details	17
3.3	Product Details	18
3.4	Product Orders	19
3.5	Complete Diagram	20
3.6	Login DFD	22
3.7	Registration DFD	23
4.1	Project Stages	24

CHAPTER 1

INTRODUCTION

This project is a web-based clone of Amazon shopping system for an existing ecommerce website i.e. Amazon. The project objective is to deliver the online shopping application into android platform.

Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. 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 computer/android device. Thus, the customer will get the service of online shopping and home delivery from his favourite shop.

1.1 PROJECT OBJECTIVE

The objective of the project is to make an application in website/android platform to purchase items in an amazon clone shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view.

1.2 PROJECT OVER VIEW

The Amazon Clone Services department strives to provide solutions to develop and transfer easy and efficient way in the digital age and to help reduces the human pressure and time. To help support shop collections, the digital initiatives, and external partner institution digital projects, it provides services that include the digitization of analogue objects, metadata management, digital preservation, and discovery and access of digital collections. Amazon Clone is web application written for all operating system, designed to help users maintain and organize shop virtually. This application is easy to use for both beginners and advanced

users. It features a familiar and well thought-out, an attractive user interface, combined with strong searching insertion and reporting capabilities. The report generation facility of shop system helps to get a good idea of which are the various items brought by the members, makes users possible to get the product easily.

1.3 PROJECT SCOPE

This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won9t be losing any more customers to the trending online shops such as Amazon itself. Since the application is available in the web browser/Smartphone it is easily accessible and always available.

1.4 STUDY OF THE SYSTEM

1.4.1 MODULES

The system after careful analysis has been identified to be presented with the following modules and roles.

The Modules involved are:

1.4.1.1 USERS

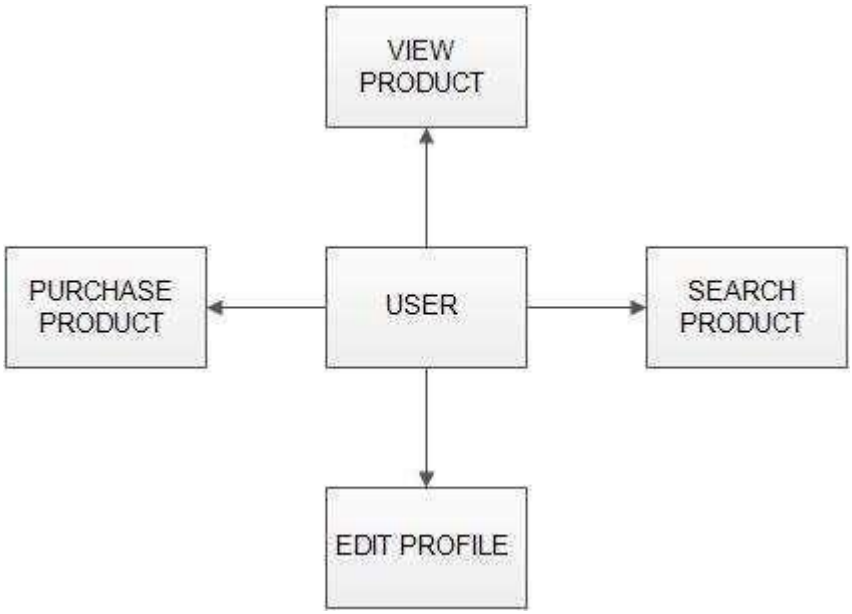


Fig 1.1 User Module

- **Registration**

  A new user will have to register in the system by providing essential details in order to view the products in the system, The admin must accept a new user by unblocking him.

- **Login**

  A user must login with his/her username and password to the system after registration.

- **View Products**

  User can view the list of products based on their names after successful login. A detailed description of a particular product with product name, products details, product image, price can be viewed by users.

- **Search Product**

  Users can search for a particular product in the list by name.

- **Add to cart**

  The user can add the desired product into his/her cart by clicking add to cart option on the product. He/she can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

- **Submit Cart**

  After confirming the items in the cart user can submit the cart providing a delivery address. On successful submitting the cart will become empty.

- **History**

  In the history the user will have a view of pending orders.

CHAPTER 2

REQUIREMENT AND SYSTEM ANALYSIS

Requirement & 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.

2.1 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.

a) It is less user-friendly.
b) User must go to shop and select products.
c) It is difficult to identify the required product.
d) Description of the product limited.
e) It is a time-consuming process.
f) Not in reach of distant users.

2.2 PROPOSED SYSTEM

In the proposed system customer need not go to the shop for buying the products. He can order the product he wishes to buy through the application in his browser /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.

2.3 SYSTEM REQUIREMENT SPECIFICATION

2.3.1 GENERAL DESCRIPTION

Product Description:

The system consists of two parts. A web application which can provide the online shopping service and an application for the customer to Online Shopping System access the web service from his Smartphone. Web application should be able to help the customer for selecting his item and to help the owner in managing the orders from the customers.

Problem Statement:

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands, if shops are providing an online Portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

2.3.2 SYSTEM OBJECTIVES

- To provide a web application for online shopping of products in an existing shop.
- To provide a online shopping web site for the same shop.

2.3.3 SYSTEM REQUIREMENTS

2.3.3.1 NON-FUNCTIONAL REQUIREMENTS

I. EFFICIENCY REQUIREMENT

When an online shopping cart web application implemented customer can purchase product in an efficient manner.

II. RELIABILITY REQUIREMENT

The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

III. USABILITY REQUIREMENT

The web application is designed for user friendly environment and ease of use.

IV. IMPLEMENTATION REQUIREMENT

Implementation of the system using CSS and HTML in front end with firebase as back end and it will be used for database connectivity. And the database part is developed by Google firebase. Responsive web designing is used for making the website compatible for any type of screen.

2.3.3.2 FUNCTIONAL REQUIREMENTS

USER

- USER LOGIN

Description of feature

This feature used by the user to login into system. A user must login with his username and password to the system after registration. If they are invalid, the user not allowed to enter the system.

Functional requirement

Username and password will be provided after user registration is confirmed. - Password should be hidden from others while typing it in the field.

- REGISTER NEW USER

Description of feature

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

Functional requirement

- System must be able to verify and validate information
- The system must encrypt the password of the customer to provide security.

- PURCHASING AN ITEM

Description of feature

The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart, the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

Functional requirement

- System must ensure that, only a registered customer can purchase items.

2.3.3.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Required

- Processor: 11th Gen Intel® Core ™ or Above
- RAM: 2GB or Above
- Hard Disk: 4GB or Above
- Input Devices: Keyboard, Mouse
- Output Devices: Monitor

Software Required

- Operating System: Linux, Ubuntu, Mac, Windows XP, 7, 8, 8.1, 10
- Frontend: HTML, CSS, Javascript, ReactJS
- Backend: Firebase
- Local Host: local host 3000
- IDE: Visual Studio Code or any other IDE's

2.4 FEASIBILITY ANALYSIS

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

Technical Feasibility

It is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well.
All the resources needed for the development of the software as well as the maintenance.

Economic Feasibility

Development of this application is highly economically feasible. The organization needed not spend much one for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even afterthe development, the organization will not be in a condition to invest more in the organization. Therefore, the system is economically feasible.

2.5 PLANNING AND SCHEDULING

A Gantt chart is a commonly used graphical depiction of a project schedule. It's a type of bar chart showing the start and finish dates of a project's elements such as resources, planning

and dependencies.

a) A Gantt chart is a visualization that helps in scheduling, managing and monitoringspecific tasks and resources in a project.

b) It consists of a list of tasks and bars depicting each task's progress.

c) It's the most widely used chart in project management.

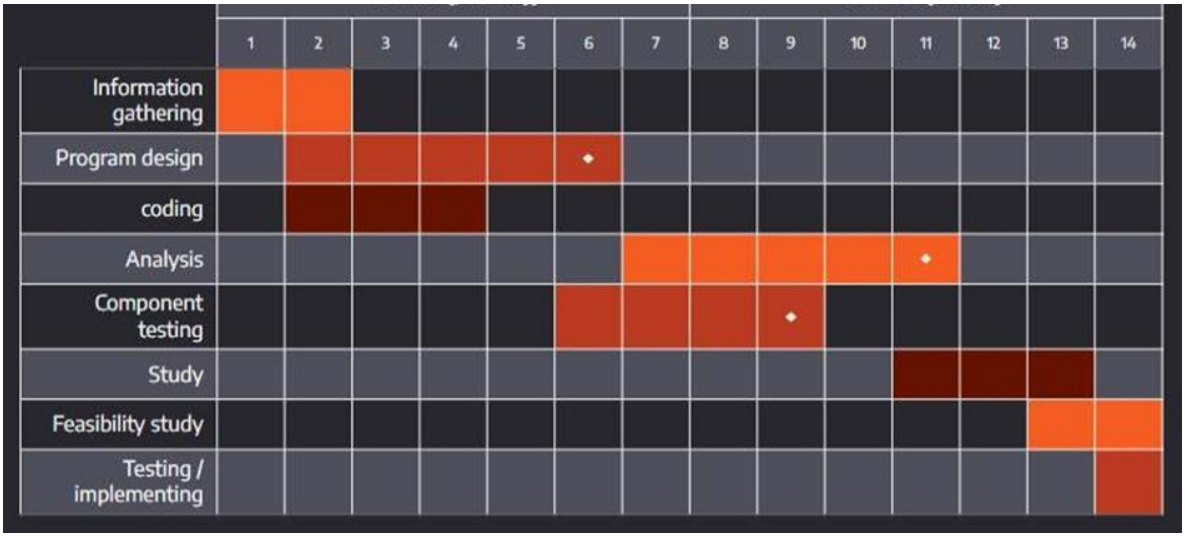


Fig 2.1 Gantt Chart for Amazon Clone Project

2.6 CONTEXT DESIGN

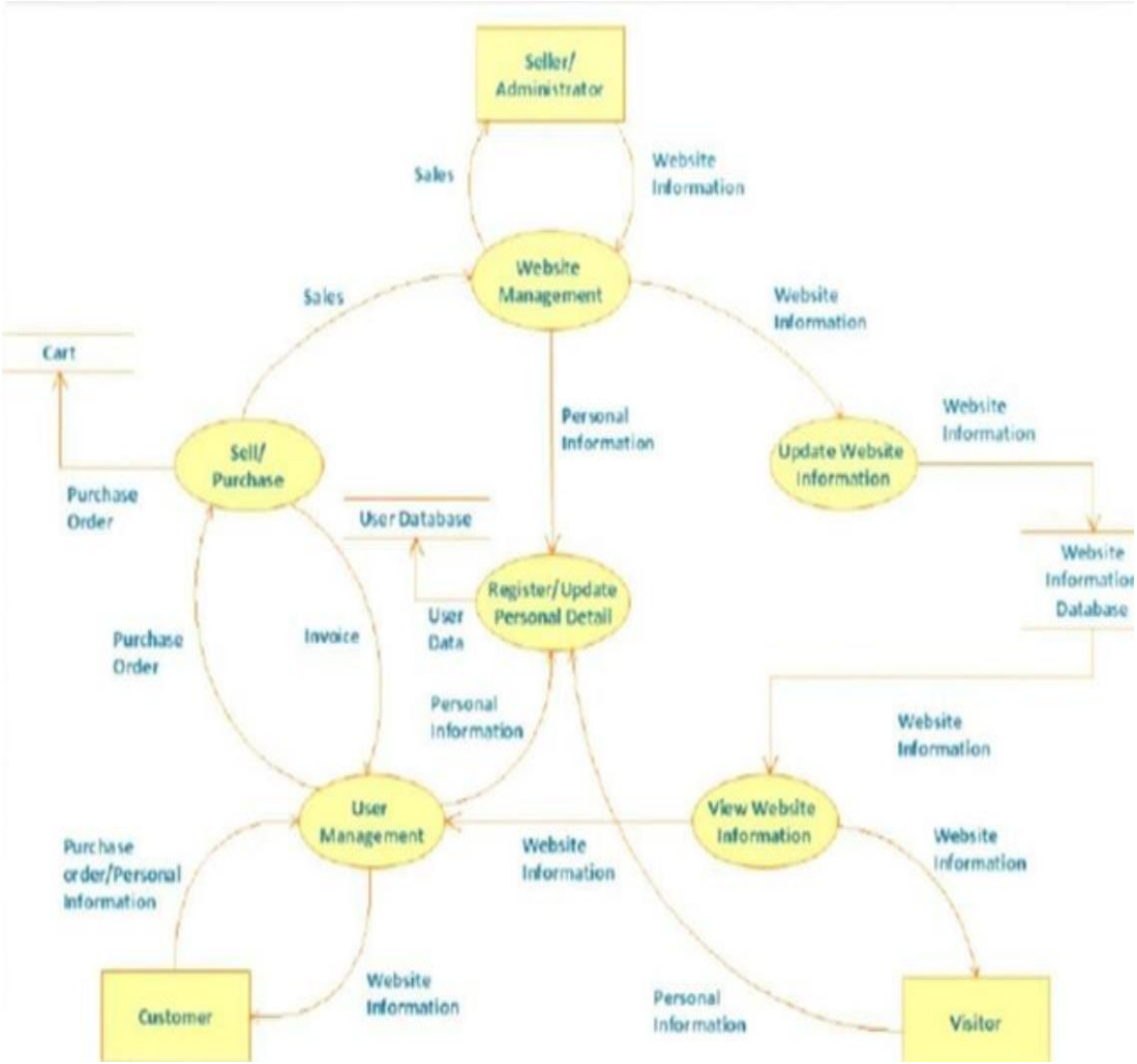


Fig 2.2 Context Design

CHAPTER 3

DESIGN OF THE SYSTEM

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

- Logical design
- 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.

3.1 INPUT AND OUTPUT DESIGN

3.1.1 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.

3.1.2 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.

3.2 DATABASE

DATABASE DESIGN

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

3.3 SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

3.3.1 FRONT END

HTML, CSS, JAVA SCRIPT, REACTJS, VISUAL STUDIO CODE is utilized to implement the frontend.

REACTJS

React is a JavaScript-based UI development library. Facebook and an open-source developer community run it. Although ReactJs is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development. React offer various extensions for entire application architectural support, such as Flux and React Native, beyond more UI.

HTML (Hyper Text Mark-Up Language)

HTML is a syntax used to format a text document on the web.

CSS (Cascading Style Sheets)

CSS is a style sheet language used for describing the look and formatting of a document written in a mark-up language.

JAVASCRIPT

JS is a dynamic computer programming language. It is most commonly used as part of webbrowsers, whose implementations allow client-side scripts to interact with the user, controlthe browser, communicate asynchronously, and alter the document content that is displayed. JavaScript is used to create popup windows displaying different alerts in the system like User registered successfully, Product added to cart etc.

VISUAL STUDIO CODE

The application is delivered to customer through a browser application. So, VS CODE platform is used to develop the user application.

3.3.2 BACK END

The back end is implemented using GOOGLE Firebase which is used to design the databases.

FIREBASE

Firebase is a toolset to build, improve, and grow your app, and the tools it gives you cover a large portion of the services that developers would normally have to build themselves, but don9t really want to build, because they'd rather be focusing on the app experience itself. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud, and scale with little to no effort on the part of the developer.

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google9s infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

3.5 ER DIAGRAMS

- LOGIN

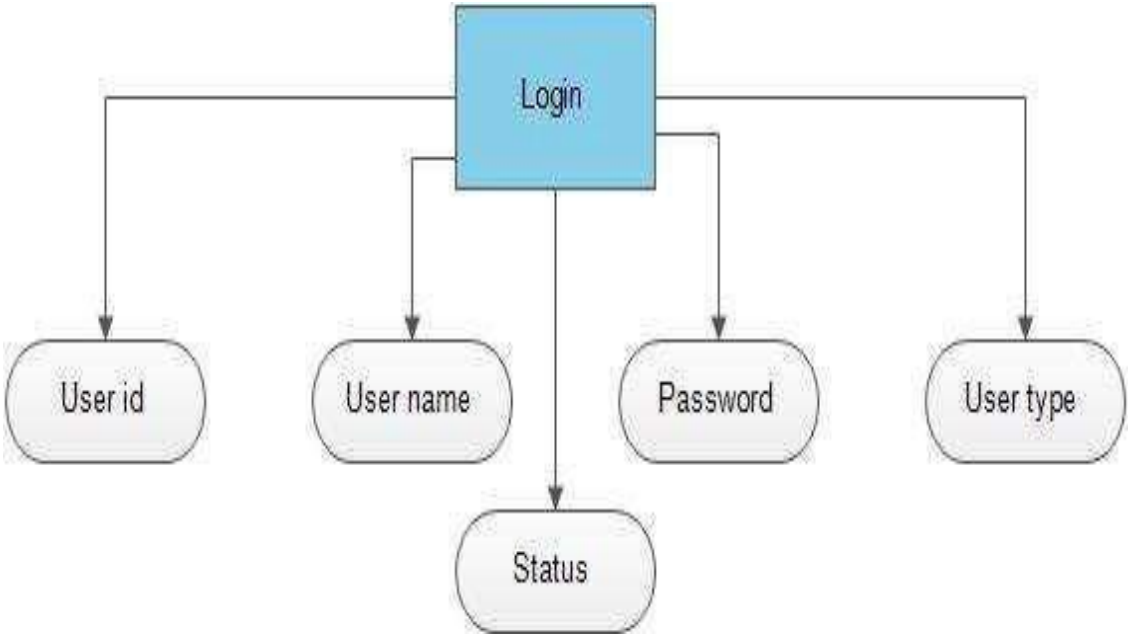


Fig 3.1 Login

- USER DETAILS

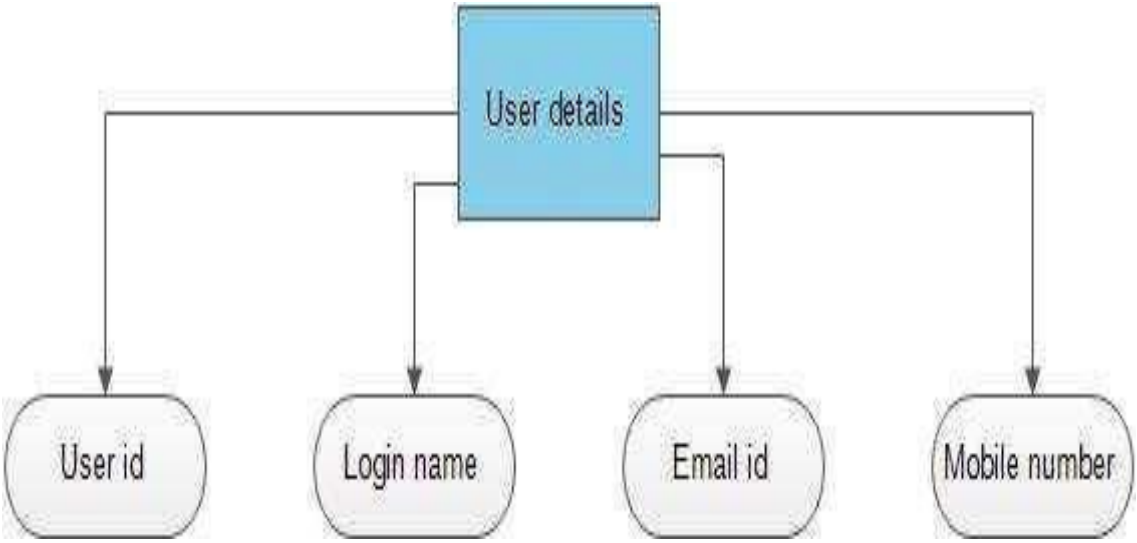


Fig 3.2 User Details

- PRODUCT DETAILS

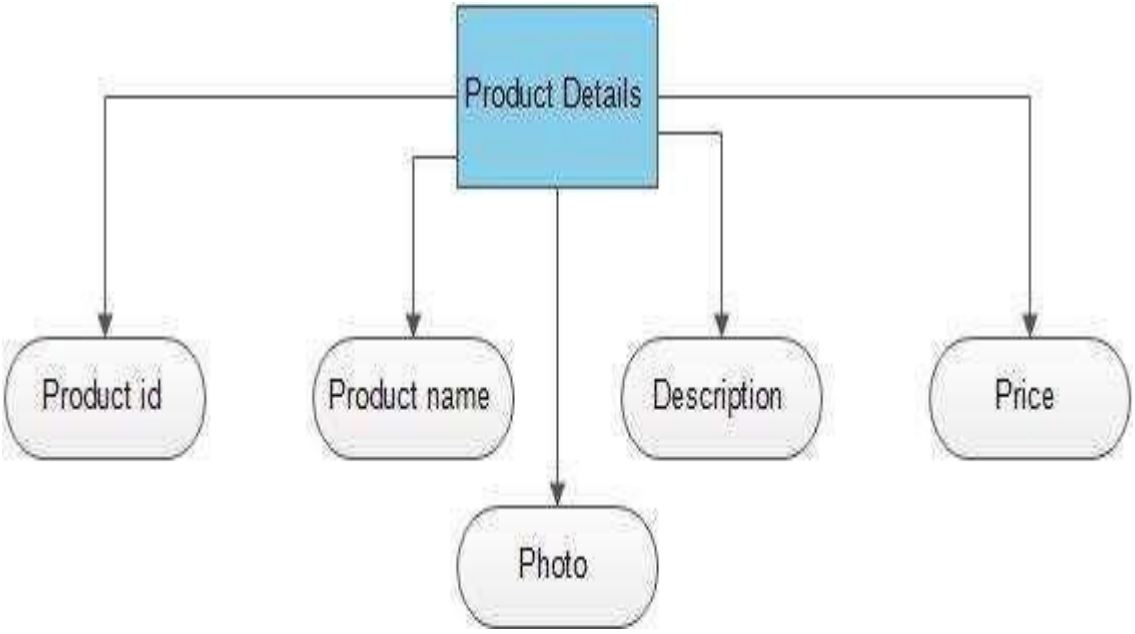


Fig 3.3 Product Details

- PRODUCT ORDERS

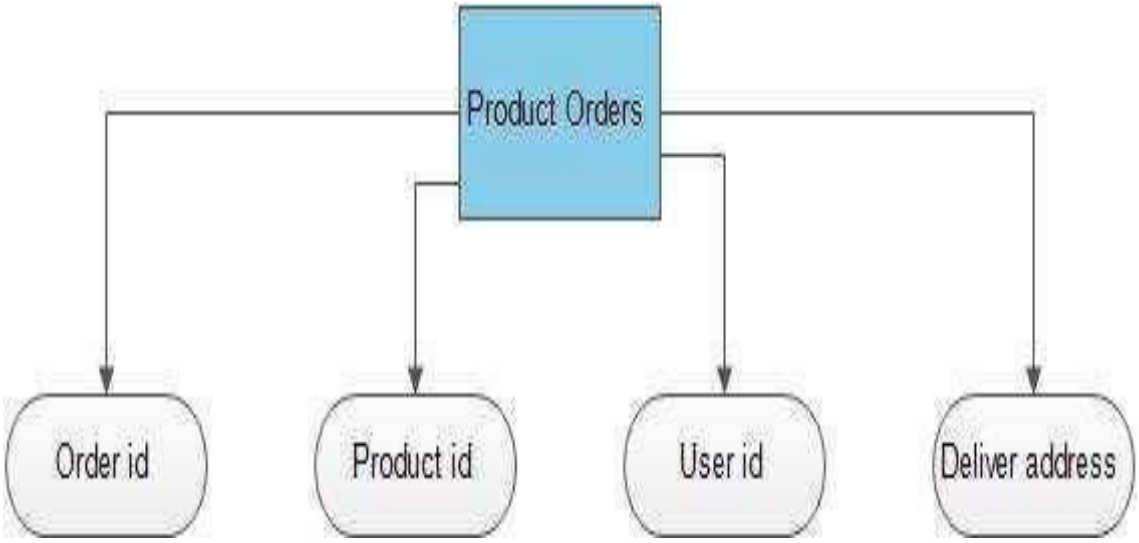


Fig 3.4 Product Orders

- COMPLETE DIAGRAM

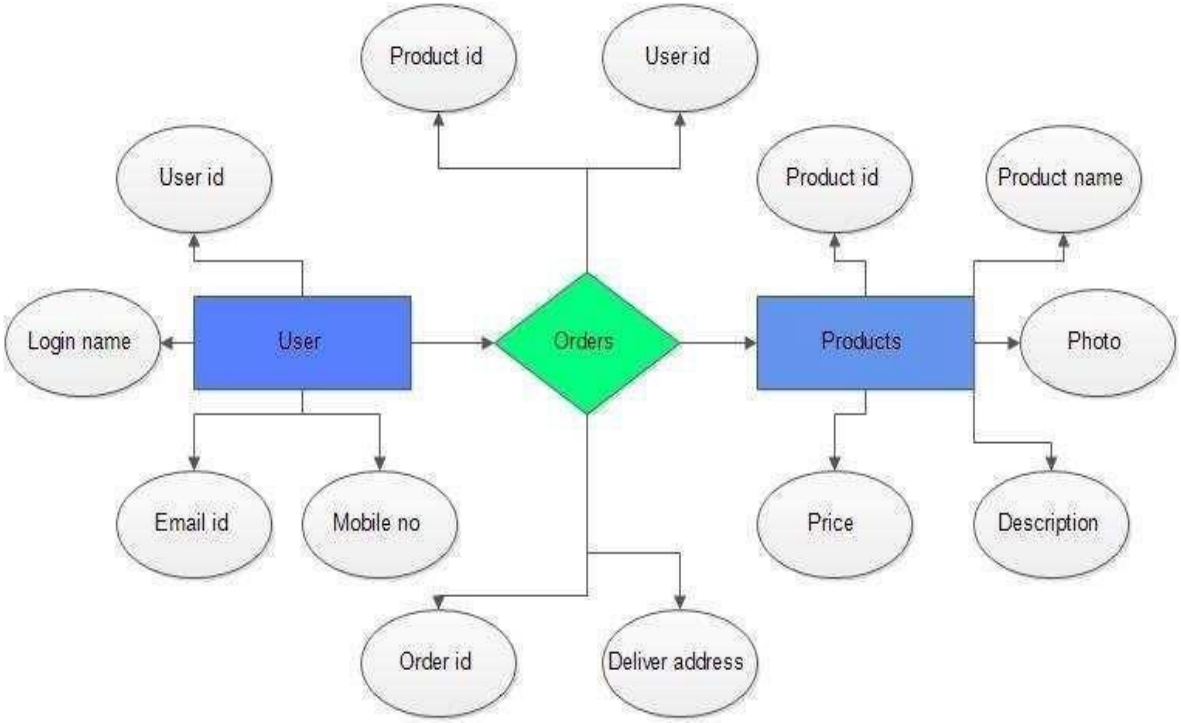


Fig 3.5 Complete Diagram

3.6 DATA FLOW DIAGRAM

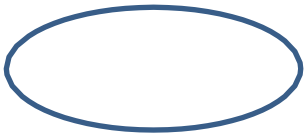
A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any 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 DFD9s 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.

PROCESS

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



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 process and 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 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.

- LOGIN DFD

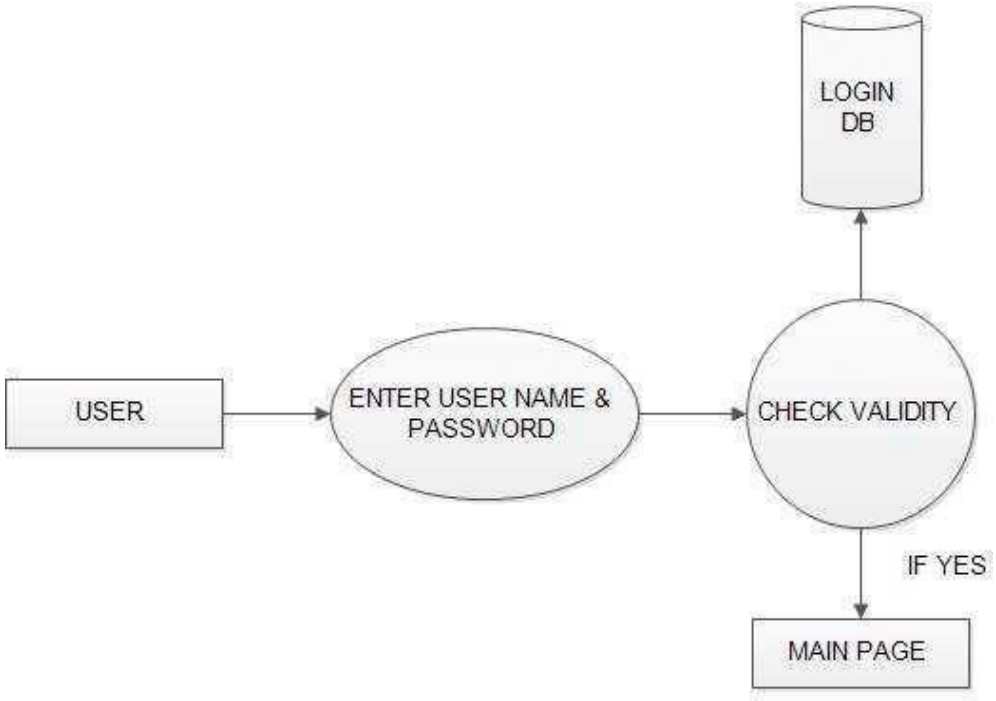


Fig 3.6 Login DFD

- REGISTRATION DFD

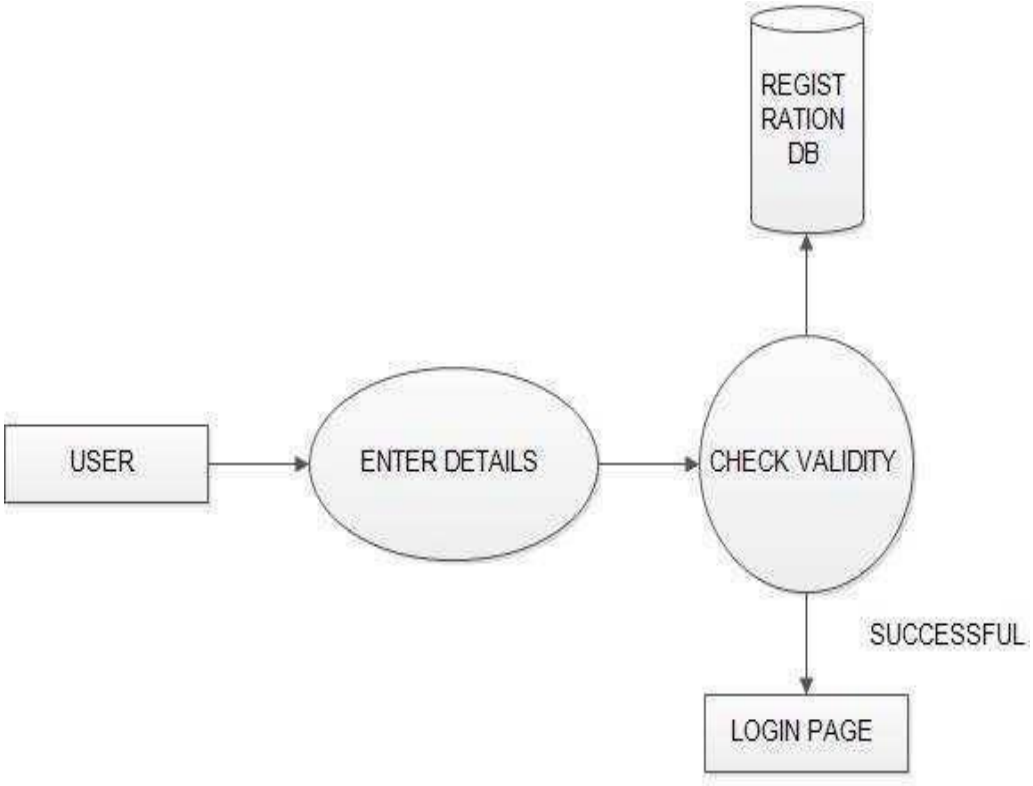


Fig 3.7 Registration DFD

CHAPTER 4

METHODOLOGY

4.1 PROJECT STAGES

The project consists of the following stages:

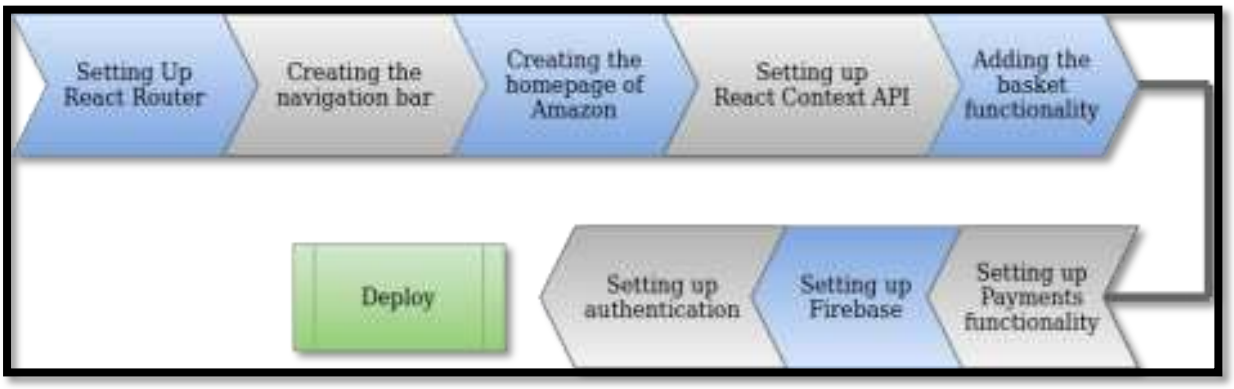


Fig 4.1 Project Stages

4.2 HIGH-LEVEL APPROACH

a) The first task, once we get the development environment ready, will be to set up the React Router.
b) Once we have everything in place, we can start off with creating the website header, which will basically serve as the navigation bar, as in most modern websites.
c) Next up is the home page building. In this project, we will be keeping it simple by showing all our sample products on the homepage.
d) Then we will be setting up the React Context API. The Context API is a component structure provided by the **React** framework, which enables us to share specific states across all levels of the application. In our project, we9ll need to manage two states:

basket (to manage the shopping cart) and user (for managing the details of the currently logged in user).

e) For setting up the payment's functionality, we'll be using APIs provided by Stripe. Handling our database and authentication needs to be supported and we'll be using Firebase for the same. Basically, the database will be used to store the login information for the users, but the resource can be used for storing product information as well.

f) Once we have Firebase setup, we can work on the Login page of our application. Successful implementation of the above requirements will lead to completion of the core implementation of our e-commerce solution.

4.3 PRIMARY GOALS

- Create header/navigation bar to navigate between pages.
- Create a home page to display products.
- Create a login page for user login.
- Add functionalities like basket, payment and authentication.

4.4 Environment Setup

Before the start of any development procedure, we need to set up the environment according to our application needs.

Requirements

a) Install Node.js on your machine.

b) Install and set up a React application.

c) Start the development server to verify prior installation.

d) Let's get started as we create a new project using the create-react-app so go to a directory where you will store this project and type the following in the terminal.

create-react-app amazon-clone
Bash

e) The above command uses the create-react-app CLI tool to generate a react boiler plate project for us so that we don't have to configure any tooling.

f) For the command above to work, the create-react-app CLI tool must be installed globallyusing the command below.

npm install -g create-react-app
Bash

g) Spin up the server with the following:

cd amazon-clone
npm start

h) Like any typical application, the source code should be in a src folder. Delete the 3 optional files from that folder: App.test.js, logo.svg and setup Tests.js. You'll not need them.

i) Remove all the content of the App.css file.

j) Remove the unnecessary code from the App() function of App.js.

4.5 Navigation with the React Router

- React is a single page application. Which means that it doesn't support multiple routes by default? Routing is the ability to move between different parts of an application when a user enters a URL or clicks an element (link, button, icon, image etc.) within the application. It enables us to transition from one view to another.
- As mentioned earlier, moving between pages is one of the most salient features to be taken care of. So, in order to enable navigation in our application, we'll take the aid of a node package named **react-router-dom**.

Requirements

- Install React Router.
- Make a new component called Home. **Components** are independent and reusable bits of code. They serve the same purpose as JavaScript functions but work in isolation and return HTML via a render function. Traditionally, every component hasa .js file and a .css file. Follow the **BEM** (Block Element Modifier) convention when naming your components. We9ll be working on functional components throughout this project. In simple words, functional components are JavaScript functions. By writing a JavaScript function, we can create a functional component in React Apps.

4.6 Creating the navigation bar

Now that we have our development environment and the React router in place, it's time to begin the actual job! We're going to start off by building the navigation bar of our web app.

Requirements

- We need a couple of icons in the navigation bar, like the search icon and the basket icon. React has a UI framework called **Material-UI**, which helps in building our frontend. Install the @material-ui/core and the @material-ui/iconspackages.
- Create a new component called Header. As mentioned earlier, we'll be working on functional components throughout this project. Fulfil the following requirements:

- Add the amazon logo to the navigation bar.
- Add a search bar. The search bar should also have a search icon. It need not be functional at the moment.
- Create the navigation section of the header. Add the following sections:
    - Login
    - Returns & Orders
    - Amazon Prime
    - Shopping Basket
- Add the necessary code in the .css file of the component so that it looks similar to the one of Amazon.
- Include only the Header component in the App.js and use it at Route path = "/".

4.7 Creating the homepage

It's time for us to work on the homepage. The homepage of any website serves as the default page of that website. That is reason enough for the homepage to be really expensive as well as creative.

Requirements

- Go to your Home.js, i.e.,
- Create a Product component. As mentioned before, you9ll need two files: Product.js and Product.css. To attain reusability, pass parameters (referred to as props in React.js) like id, title, image, price and rating to your functional component.
- Utilize the Product component in your home component to render out a few sample products. Use proper flex CSS styles where required to get the desired (similar to Amazon.com products) result.

4.8 Setting Up React Context API

The Context API provides a way to share data values between components without having to pass a prop through every level of the app tree. The React Context API is a component structure provided by the React framework. It basically solves the problem of prop drilling.

Requirements

- Create a file StateProvider.js
- Create a file reducer.js which fulfils the following criteria.
- Declare all application-level states which are to be used later by the application.
- Create a function which can calculate the total amount of the items in theshopping basket.
- Export the reducer.
- Do the following in index.js
- Import the reducer and initial State from reducer.js and StateProvider component from StateProvider.js.
- Enclose the App component with StateProvider so that the child's components can get access to the states too.
- Pass the reducer and initial State to the StateProvider.

4.9 Adding the Basket Functionality

- In your Product.js, import the useStateValue hook from the StateProvider.
- You need to use the useStateValue hook to bring in a state named basket (for example) andalso get a dispatch function which allows you to dispatch actions to change the state in the reducer.
- Import the useStateProvider to your Header component and make use of the basket information to show the number of items in the basket.

4.10 Setting Up Firebase and Authentication

Firebase is a great service provided by Google for configuring the backend of any web application with all the general necessities like database preparation, authentication using various methods, etc.

Requirements

- Create a project on firebase.
- Setup the database and setup sign-in method using Email/Password.
- Register your application and set up firebase hosting.
- Install necessary firebase dependencies in your local setup.
- Open the firebase SDK snippet section in your firebase project and copy paste the necessary configuration in a file named firebase.js.

4.11 The Checkout Page

- Create a Checkout component on the Checkout page.
- The Checkout component renders this page and will use two more components.

4.12 Payments Functionality and more

By now our application looks good and is a proper e-commerce solution, except it is missing one very important aspect of commerce and that is the payment facility. Moreover, this application is a clone inspired from Amazon.com. We can add more functionality and tickle with the styling of the application to make it our own.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 IMPLEMENTATION APPROACHES

Plan of Implementation

- Research
- References projects
- Designing UI
- Coding Logic
- Creating Tables in DB
- Joining database to main
- Bugs fixing

5.2 CODE EFFICIENCY

CODE EFFICIENCY

The code is efficient and does it's work properly according to the needs.But it also contains some minor bugs. The bugs are as follows:

- The Final amount of the PRODUCT does not increase even if number of items are added.
- Sometimes we are not able to see the data in the <Data table" due to connection errorswith the database.

5.3 TESTING APPROACH

- Tool-bars work properly? Is all menu function and pull-down sub function properly listed? Is it possible to invoke each menu function using logical assumptions that if all parts of the system are correct, the goal will be successfully achieved? In adequate testing or non-testing will leads to errors that may appear few months later. Testing represents an interesting anomaly for the software engineer. During earlier software

engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test cases that areintended to <demolish= the software that has been built. In fact, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive. Testing requires that the developer discard preconceived notions of the <correctness= of software just developed and overcome a conflict of interest that occurs when errors are uncovered.

- If testing is conducted successfully (according to the objectives stated previously) it will uncover errors in the software. As a secondary benefit, testing demonstrates that software functions appear to be working according tospecification, that behavioural and performance requirements appear to have been met. In addition, data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as a whole. But testing cannot show the absence of errors and defects, it can show only that software errors and defects are present. It is important to keep this (rather gloomy) statement in mind as testing is being conducted.
- There are types of testing that we implement. They are as follows:
- While deciding on the focus of testing activities, study project priorities. For example, for an on- line system, pay more attention to response time. Spend more time on the features used frequently. Decide on the effort required for testing based on the usage of the system. If the system is to be used by a large number of users, evaluate the impact on users due to a system failure before deciding on the effort. This creates two problems
    - Time delay between the cause and appearance of the problem.
    - The effect of the system errors on files and records within the system.
- The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits. The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out. There are two major types of testing

they are:

- White Box Testing.
- Black Box Testing.

5.3.1 WHITE BOX TESTING

White box sometimes called <Glass box testing= is a test case design uses the control structure of the procedural design to drive test case. Using white box testing methods, the following tests were made on the system

All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed.

All logical decisions were checked for the truth and falsity of the values.

5.3.2 BLACK BOX TESTING

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like.

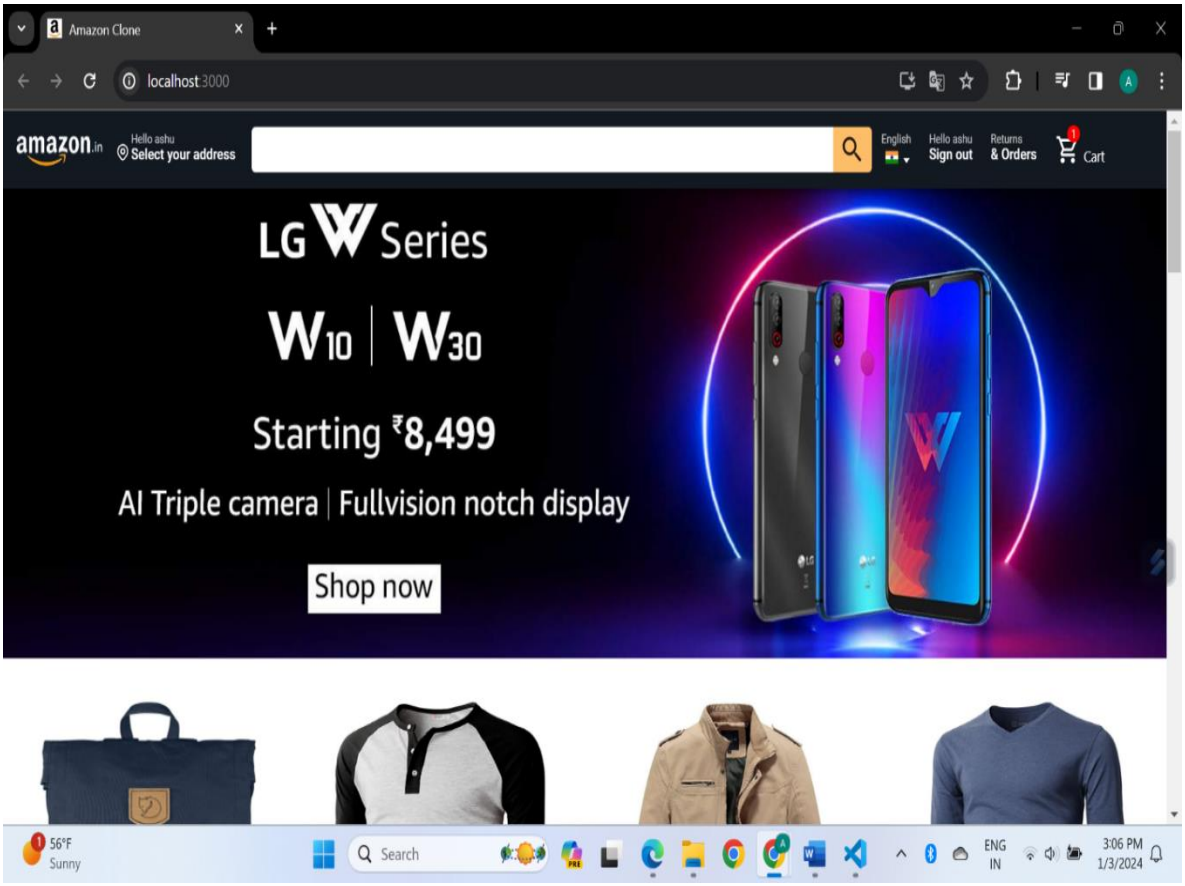
- Interface errors.
- Performance in data structure.
- Performance errors.
- Initializing and termination errors.

CHAPTER 6

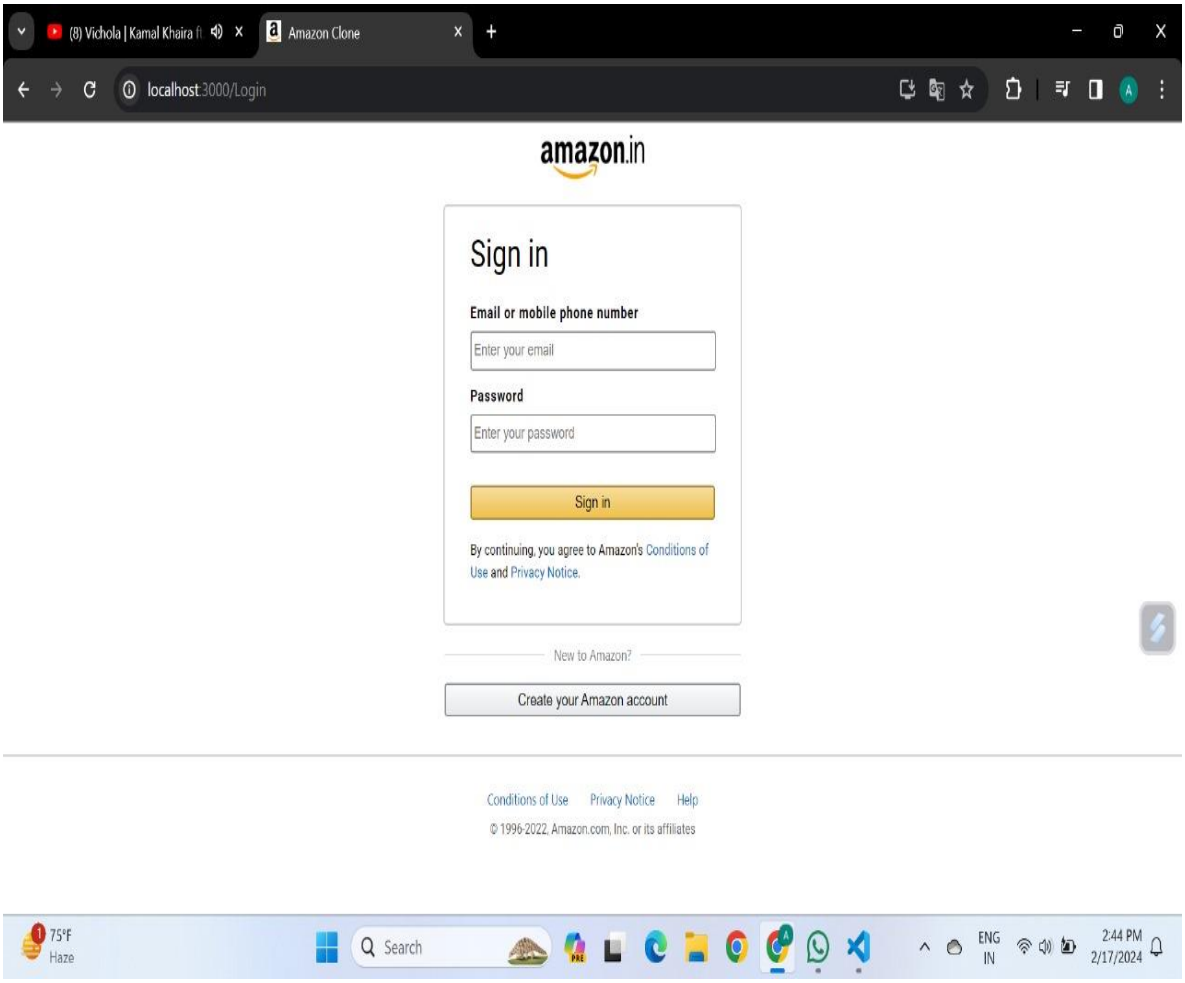
RESULT

6.1 SCREENSHOTS

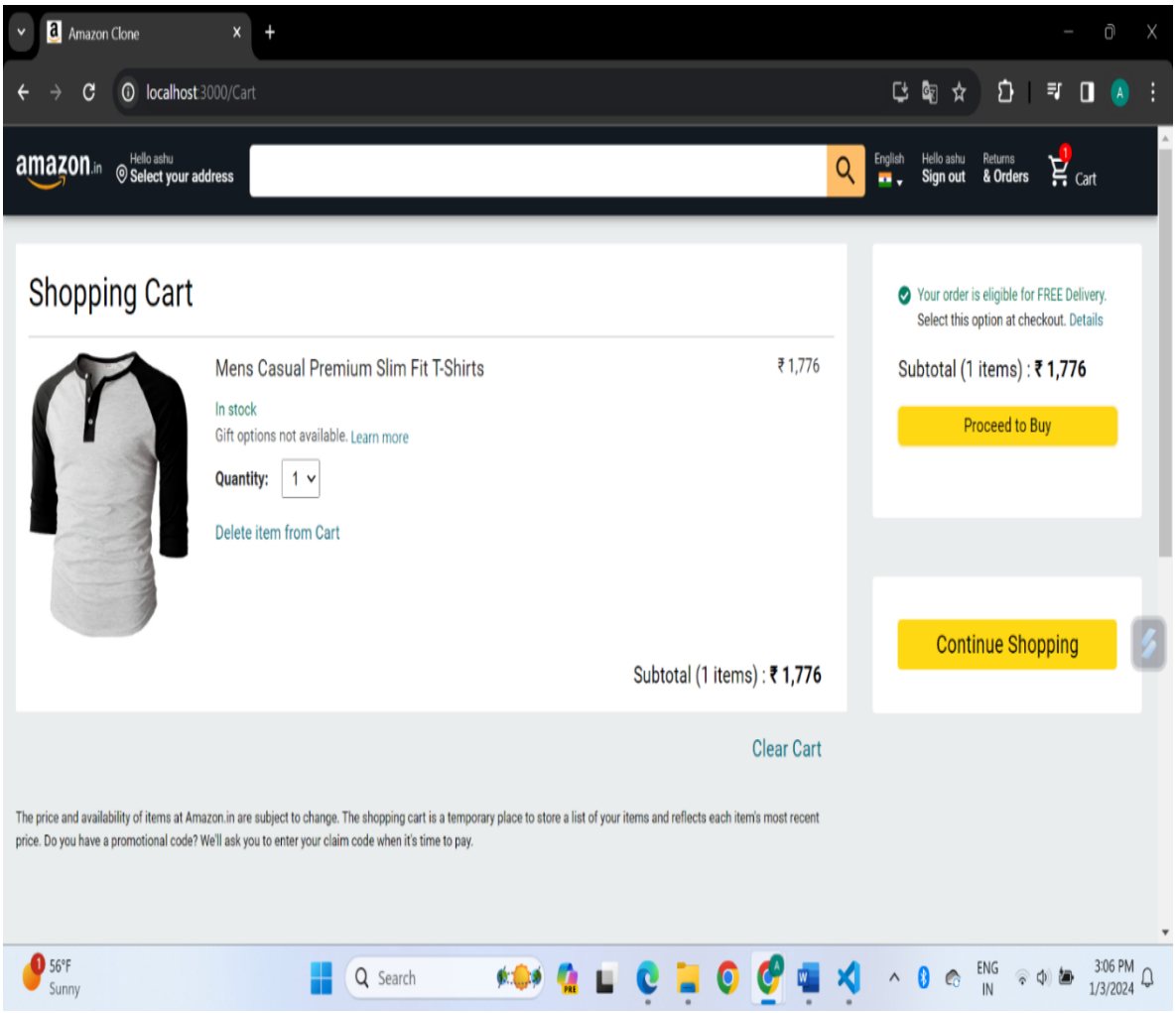
6.1.1 HOME PAGE



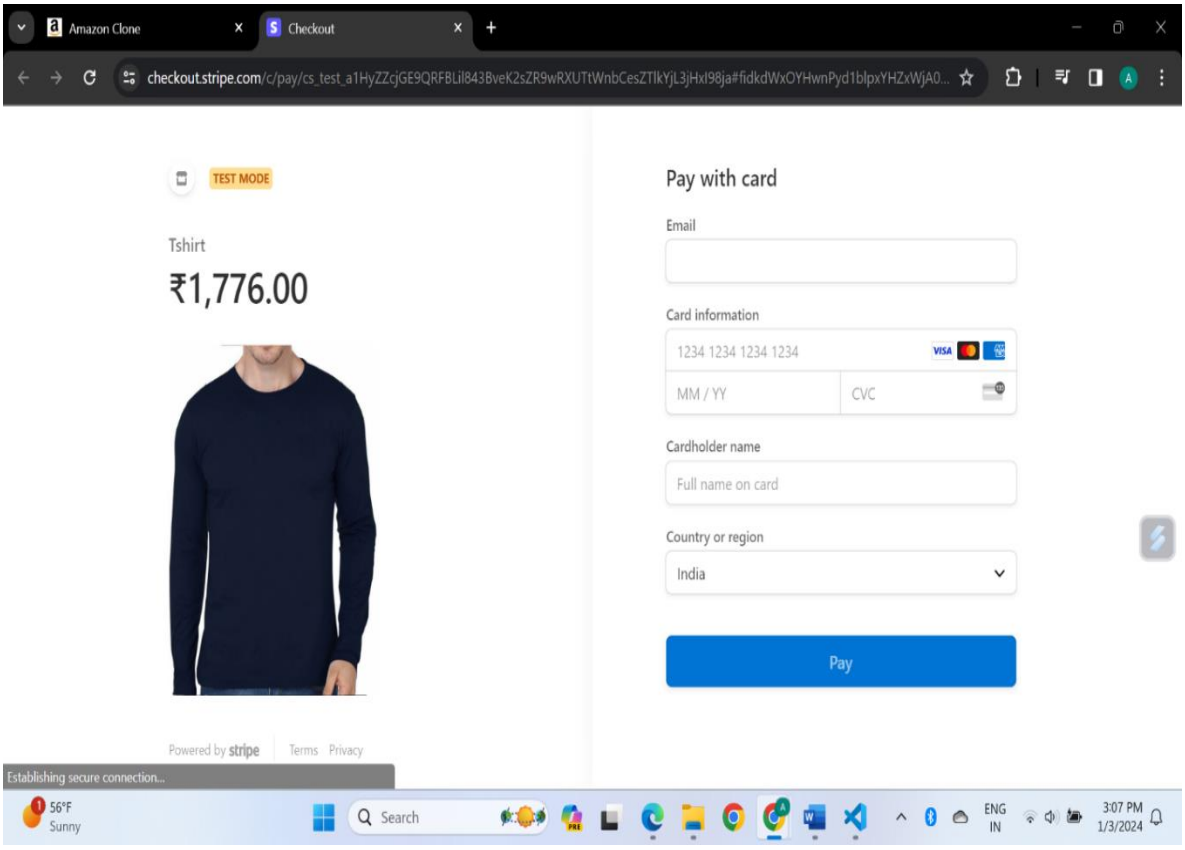
6.1.2 SIGN IN PAGE



6.1.3 SHOPPING CART



6.1.4 PAYMENT PAGE



CHAPTER 7
CONCLUSION

7.1 CONCLUSION

The project entitled Amazon clone website was completedSuccessfully.
The world has moved online – a fact that businesses have to accept and put up a website to address. Amazon is a prime example of a website with all the key elements making up a good e-commerce site. The e-commerce website of Amazon was initially put together with simple HTML, CSS and JavaScript. But as time progressed and different frameworks came into the limelight, the website got a makeover.
This project we are building an e-commerce application using React, which is inspired by Amazon. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. Through this project, we9ll discover a way to build a functional clone of Amazon9s e-commerce website, relying on React and Firebase.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using HTML & CSS, usage of responsive templates, designing of web applications, and management of database using firebase. The entire system is secured. Also, the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.
This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications. There is a scope for further development in our project to a great extent.

7.2 FUTURE ENHANCEMENT

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications. There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like providing moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limit us. The following are the future scope for the project.

- Should be added payment gateway.
- Can be added inventory management system.
- Can be added multiple branches.
- Can be added multilingual to this site.
- And many features can be added this project to make it more robust.

REFERENCES

http://www.w3schools.com

http:// www.stackoverflow.com

http://wikipedia.com

https://developer.mozilla.org/en-US/

https://legacy.reactjs.org/docs/create-a-new-react-app.html

https://legacy.reactjs.org/tutorial/tutorial.html

https://react.dev/learn/your-first-component

https://firebase.google.com/docs/auth/android/start

https://blog.devgenius.io/how-to-implement-stripe-into-your-react-site-in-4-steps-715ee5b92160

https://www.youtube.com/

https://www.google.com/