A

Capstone Project Report

On

CLOUD BASED E-COMMERECE WEBSITE

Submitted to JNTU HYDERABAD

In Partial Fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted
By

A.SAI RAM **(228R1A6608)**

Under the Esteemed guidance of

Mrs. B. REVATHI

Assistant Professor, Department of CSE(AI&ML)

Department of Computer Science and Engineering
(AI&ML)

CMR ENGINEERING COLLEGE
(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

(Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

(2024-2025)

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE NEW DELHI, Affiliated to JNTU, Hyderabad)

(Kandlakoya, Medchal Road, R.R. Dist. Hyderabad-501 401)

Department of Computer Science and Engineering (AI&ML)

CERTIFICATE

This is to certify that the project entitled **"CLOUD BASED E-COMMERCE WEBSITE"** is a bonafide work carried out by

A.SAI RAM **(228R1A6608)**

in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING (AI&ML)** from CMR Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision.

The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide
Mrs. B. REVATHI
Assistant Professor
Department of CSE(AI&ML)
CMREC, Hyderabad

Head of the Department
Dr. P. MADHAVI
Professor & HOD
Department of CSE(AI&ML)
CMREC, Hyderabad

DECLARATION

This is to certify that the work reported in the present project entitled **"CLOUD BASED E-COMMERECE WEBSITE"** is a record of bonafide work done by us in the Department of Information Technology, CMR Engineering College, JNTU Hyderabad. The reports are based on the project work done entirely by me and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

A.SAI RAM (228R1A6608)

ACKNOWLEDGEMENT

I am extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Madhavi Pingili**, HOD, **Department of CSE(AI&ML), CMR Engineering College** for their constant support.

I am extremely thankful to **Mrs. B. REVATHI,** Assistant Professor, Internal Guide, Department of IT, for her constant guidance, encouragement and moral support throughout the project.

I will be failing in duty if i do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

I express my thanks to all staff members and friends for all the help and co-ordination extended in bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

A.SAI RAM **(228R1A6608)**

CONTENTS

ABSTRACT

In the modern digital economy, eCommerce platforms must deliver seamless, scalable, and secure online shopping experiences. This project presents the development of a cloud-based eCommerce website hosted on Amazon Web Services (AWS), leveraging its suite of services to ensure high availability, performance, and cost-effectiveness. The system architecture is designed to be serverless and modular, utilizing Amazon S3 for static website hosting, API Gateway and AWS Lambda for backend logic, and Amazon RDS or DynamoDB for secure data storage. Amazon Cognito is integrated for user authentication and access control, ensuring secure user sessions. Product images and files are managed through Amazon S3, while CloudWatch provides real-time monitoring and logging. This cloud-native solution supports core eCommerce functionalities including user registration and login, product browsing, shopping cart management, and order processing. It also integrates a third-party payment gateway for secure transactions. The use of serverless technologies ensures automatic scalability with reduced operational overhead. Overall, the project demonstrates how AWS cloud services can be effectively utilized to build a scalable, secure, and cost-efficient eCommerce platform, suitable for startups and enterprises alike.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Introduction

The rapid growth of online shopping has significantly increased the demand for robust and scalable eCommerce platforms. Traditional hosting solutions often struggle to manage high traffic volumes, ensure 24/7 availability, and provide secure transactions. To address these challenges, cloud computing has emerged as a powerful solution that offers on-demand resources, automatic scalability, and enhanced security.

This project focuses on building a cloud-based eCommerce website using Amazon Web Services (AWS), a leading cloud service provider. By leveraging AWS, businesses can deploy flexible and reliable online stores without investing heavily in physical infrastructure. The proposed system provides essential eCommerce functionalities such as product catalog management, user authentication, shopping cart, and order processing, all built using scalable and serverless AWS components.

Key AWS services used in this project include Amazon S3 for static website hosting and image storage, API Gateway and AWS Lambda for serverless backend logic, Amazon RDS or DynamoDB for data management, and Amazon Cognito for secure user authentication. Additionally, services like CloudWatch, SNS, and CloudFront enhance the website's reliability, performance, and user experience.

This project not only highlights the advantages of cloud computing in web development but also demonstrates how modern eCommerce platforms can be efficiently implemented with minimal infrastructure management and maximum scalability using AWS.

1.2 Project Objectives

The primary objective of this project is to design and develop a fully functional cloud-based eCommerce website using Amazon Web Services (AWS) that is scalable, secure, and cost-efficient. The project aims to implement essential eCommerce features such as product browsing, user authentication, shopping cart functionality, and order management using a serverless architecture. AWS services like S3, Lambda, API Gateway, RDS/DynamoDB, and Cognito are leveraged to handle frontend hosting, backend processing, data storage, and secure user management. Additionally, the system integrates third-party payment gateways for secure transactions and uses CloudWatch for real-time monitoring. Overall, the objective is to showcase how cloud technologies can streamline the deployment and management of modern eCommerce platforms with high availability and minimal infrastructure overhead.

1.3 Purpose of the Project

The purpose of this project is to build a scalable and secure eCommerce website using AWS cloud services, allowing users to browse products, register, shop, and place orders online. By leveraging AWS's serverless and managed services, the project aims to reduce infrastructure complexity while ensuring high performance, availability, and data security.

1.4 Existing System with Disadvantages

Traditional eCommerce websites are usually hosted on physical servers or shared hosting platforms, requiring dedicated infrastructure and manual configuration. These systems handle user management, product listings, and order processing through custom backend code, often built using frameworks like PHP, Java, or .NET, and use relational databases for data storage.

Disadvantages

- Not scalable – can't handle too many users at once.
- Expensive to maintain servers and hardware.
- Downtime can occur during updates or failures.
- Security is harder to manage manually.
- Takes more time to add new features or updates.
- Difficult to connect with modern tools and services.

1.5 Proposed System with Features

The proposed system is a cloud-based eCommerce website built using Amazon Web Services (AWS), designed to be scalable, secure, and cost-effective. It uses AWS services like S3 for hosting the frontend, Lambda and API Gateway for backend logic, RDS or DynamoDB for data storage, and Cognito for secure user authentication. The system allows users to register, browse products, add items to a cart, and place orders online. It supports real-time inventory management, secure payment integration, and image storage through S3. Features like auto-scaling, high availability, and built-in monitoring with CloudWatch ensure smooth performance even during high traffic. The serverless architecture reduces maintenance and improves deployment speed, making the system efficient and easy to manage.

E-COMMERECE WEBSITE:

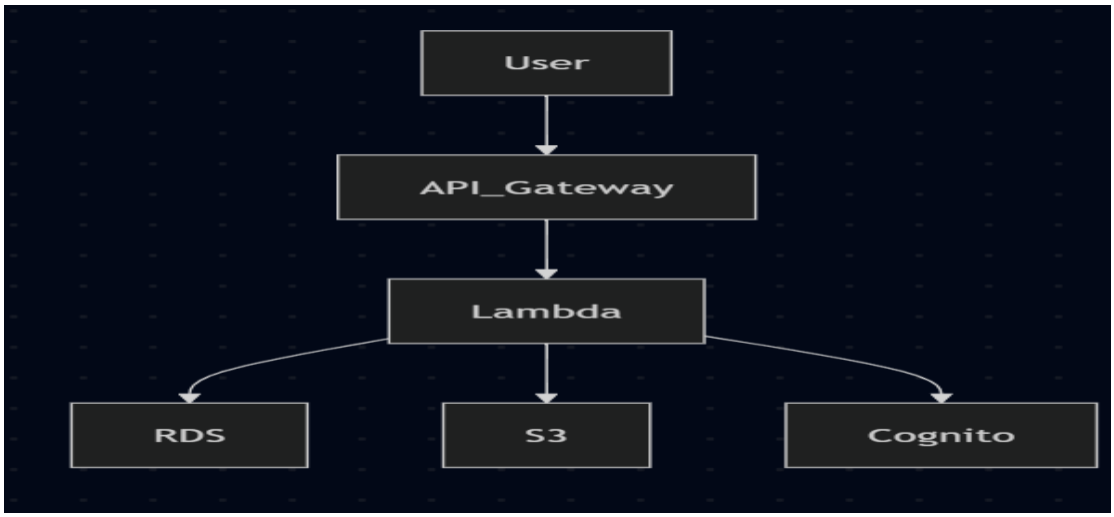

Figure 1.5.1: Block diagram of proposed system.

An eCommerce website is an online platform that enables businesses and individuals to buy and sell products or services over the internet. Unlike traditional brick-and-mortar stores, eCommerce websites provide customers with the convenience of shopping anytime and anywhere, using devices like computers, tablets, or smartphones. These websites typically feature product catalogs with detailed descriptions, images, prices, and reviews to help customers make informed purchasing decisions. Key functionalities include user registration, product search and filtering, shopping cart management, secure checkout, and order tracking. Modern eCommerce platforms also support various payment options such as credit/debit cards, digital wallets, and online banking, often integrated through secure third-party payment gateways. Additionally, they offer inventory management, customer service tools, and marketing capabilities. With the rise of cloud computing, many eCommerce websites now leverage cloud services to ensure scalability, reliability, and security. Cloud-based eCommerce platforms can easily handle large numbers of users and transactions, automatically scale during peak shopping periods, and reduce operational costs by eliminating the need for physical servers. Overall, eCommerce websites have revolutionized retail by making shopping more accessible, personalized, and efficient for both businesses and customers.

Advantages

- Open 24/7 for shopping anytime.
- Accessible from anywhere with internet.
- Saves time compared to visiting stores.
- Supports easy online payments.

1.6 Input And Output

1.7 Input Design

Input design is an essential phase in the development of an eCommerce website that focuses on designing the interfaces through which users interact with the system by entering data. It aims to create input forms and screens that are user-friendly, intuitive, and secure to ensure accurate and complete data capture. For an eCommerce platform, input design covers multiple areas such as user registration and login forms where users provide personal details and credentials; product search and filtering options to help users find items quickly; shopping cart management allowing users to add, update, or remove products easily; and checkout forms that securely collect shipping and payment information. Input validation plays a vital role in preventing incorrect or incomplete data, such as ensuring email addresses follow proper formats, mandatory fields are filled, and payment details are valid. For administrators, input design includes forms for adding and editing product information, managing stock levels, and processing orders with error checks to maintain database integrity. Well-designed input mechanisms reduce user frustration, minimize errors, enhance security, and improve overall system reliability. This careful attention to input design helps create a seamless shopping experience and ensures the backend system receives consistent and trustworthy data for processing transactions efficiently.

Objectives

- To build a secure and reliable online shopping platform where users can easily browse products, manage their accounts, and make purchases with confidence.
- To implement a seamless shopping experience by integrating features like shopping cart management, real-time inventory updates, and order tracking, ensuring customers stay informed throughout their purchase journey.
- To leverage AWS cloud services to create a scalable and highly available system that can handle varying traffic loads without compromising performance or security.
- To develop an intuitive admin interface that simplifies product management, order processing, and customer support, enabling efficient business operations.
- To ensure data privacy and security through robust authentication and secure payment integration, protecting both customer information and transaction data.

Output Design

Output design is a critical component of the eCommerce website that focuses on presenting information to users clearly, accurately, and in a timely manner. The outputs include all the visible results generated by the system in response to user actions or backend processes, such as product listings, search results, shopping cart summaries, order confirmations, and status updates. Effective output design ensures that users receive meaningful and well-organized information that enhances their shopping experience and aids decision-making.

For example, product pages should display images, prices, descriptions, and customer reviews in a clean and attractive layout. During checkout, the system must provide clear summaries of selected items, total costs, taxes, and shipping details before final order submission. After placing an order, users expect timely confirmation messages and real-time updates on shipping status or delivery tracking.

On the administrative side, the output design includes detailed sales reports, inventory status, and customer activity logs that help managers monitor and optimize business performance. Good output design also incorporates responsiveness, ensuring that all information is displayed correctly on different devices such as smartphones, tablets, and desktops.

Additionally, outputs should be accessible and user-friendly, with visual cues like buttons, notifications, and alerts that guide users through the shopping process smoothly. Overall, well-planned output design improves user satisfaction, reduces confusion, and supports efficient operation of the eCommerce platform.

2. LITERATURE SURVEY

“Indian e-commerce market to grow by 21.5% in 2022, forecasts Global Data” (Jan 21, 2022)
This report by Global Data provides a comprehensive analysis of the rapid growth trajectory of the Indian e-commerce sector, forecasting a significant 21.5% increase in market size for the year 2022. The study identifies key factors fueling this growth, such as rising internet penetration, increased smartphone adoption, improved digital payment infrastructure, and evolving consumer behavior favoring online shopping. It also discusses the expanding product categories, enhanced logistics capabilities, and government initiatives that are supporting this ecosystem. Additionally, the report highlights potential challenges like infrastructure constraints, intense competition, and regulatory hurdles, providing a balanced outlook on the opportunities and risks for businesses operating in or entering the Indian e-commerce market.

Lu Du, Xia Zhong, "Research on UGC type models of socialized e-commerce websites based on user experience" (Design, 2014)

This paper delves into the integration of user-generated content (UGC) within socialized e-commerce websites, focusing on how the design of UGC models impacts user experience and engagement. It explores how interactive elements such as reviews, ratings, comments, and social sharing can foster a sense of community and trust among consumers. The authors argue that well-structured UGC models not only enhance the credibility of products but also influence purchase decisions by leveraging social proof. By analyzing different types of UGC frameworks, the paper provides insights into optimizing social interactions on e-commerce platforms to boost user participation, satisfaction, and ultimately, sales performance.

Chaudhury, A. (2002), "e-Business and E-Commerce Infrastructure Technologies Support the e-Business Initiative"

In this study, Chaudhury presents an in-depth examination of the technological infrastructure necessary for supporting effective e-business and e-commerce operations. The paper emphasizes the importance of a robust IT framework, including networking technologies, database management systems, security protocols, and middleware, to facilitate seamless communication between businesses and customers. It highlights how these components collectively support critical functions such as transaction processing, data storage, authentication, and real-time updates. The author also discusses the evolving trends in infrastructure, such as cloud computing and service-oriented architectures, which enhance scalability, flexibility, and cost-efficiency for e-business initiatives.

Gupta, A. (2014), "E-Commerce: Role Of E-Commerce In Today's Business" (International Journal of Computing and Corporate Research)

Gupta's article provides a detailed overview of the pivotal role e-commerce plays in transforming modern business landscapes. It explores how e-commerce platforms enable companies to expand their reach beyond geographical boundaries, access new customer segments, and operate with reduced overhead costs compared to traditional retail. The paper discusses different e-commerce models, including B2B, B2C, C2C, and highlights the strategic benefits such as enhanced customer convenience, personalized marketing, and faster transaction cycles.

Le Shen, "Research on E-commerce Website Design Based on User Experience—Taking Online Digital Printing as an Example" (East China University of Science and Technology, 2013)

Le Shen's research focuses on the critical influence of user experience (UX) in the design of e-commerce websites, using the niche of online digital printing services as a practical case study. The study analyzes how website elements such as layout, navigation, visual aesthetics, and interactive features contribute to a smooth and satisfying user journey. It highlights the importance of customizing user interfaces to accommodate specific industry requirements, such as product configurators and preview tools in digital printing. The paper argues that prioritizing UX design not only attracts and retains customers but also encourages repeat purchases and positive word-of-mouth, thereby driving business growth in competitive online markets.

Amazon hosting services (https://aws.amazon.com/what-is/web-hosting/)

This official AWS resource offers an extensive overview of Amazon Web Services' web hosting solutions, demonstrating how AWS enables businesses to deploy, manage, and scale websites and applications in the cloud. It outlines various hosting models, from static website hosting using Amazon S3 to dynamic hosting on EC2 virtual servers and fully serverless architectures with AWS Lambda. The page highlights AWS's strengths in providing high availability, scalability, security, and cost optimization, catering to businesses of all sizes and technical requirements. It also discusses integrated services such as Amazon CloudFront for content delivery, Route 53 for DNS management, and AWS Certificate Manager for SSL security, positioning AWS as a comprehensive platform for modern web hosting needs.

3. SOFTWARE REQUREIMENTS ANALYSIS

3.1 Problem Statement

In today's digital era, consumers increasingly prefer online shopping due to its convenience, variety, and accessibility. However, many existing eCommerce platforms face challenges such as limited scalability, poor performance during high traffic periods, lack of data security, and high operational costs. Traditional hosting solutions often require complex infrastructure management and are not cost-effective for small to medium-sized businesses. There is a need for a robust, secure, and scalable eCommerce solution that can adapt to fluctuating demands and ensure seamless user experience. This project aims to address these challenges by designing and deploying a cloud-based eCommerce website using Amazon Web Services (AWS), leveraging serverless technologies and cloud-native services to provide high availability, improved performance, reduced downtime, and cost-efficiency.

3.2 Modules and Their Functionalities

Data Preprocessing

The cloud-based eCommerce website is built using a modular architecture to ensure scalability, maintainability, and performance. The core modules include the User Management Module, which handles user registration, authentication, and profile management using AWS Cognito. The Product Management Module enables administrators to manage product listings, while users can browse through the product catalog efficiently. The Search and Filter Module enhances user experience by allowing easy discovery of products through keywords and categories. Users can add items to their cart via the Shopping Cart Module, and place orders using the Order Management Module, which also tracks order history and generates invoices. Secure transactions are supported through the Payment Gateway Integration Module, which integrates with external payment services. The Admin Dashboard Module provides real-time insights into users, sales, and inventory, while the Notification Module uses AWS services to send emails and alerts for order confirmations and promotions. Additionally, an optional Recommendation System Module can be integrated to suggest personalized products using machine learning. This refund data is then used for analytics and model training, enabling personalized experiences and data-driven decision-making across the platform.

3.3 Non-Functional Requirements

The non-functional requirements of the cloud-based eCommerce website are essential to ensure optimal performance, scalability, and user satisfaction. Scalability is a key requirement, allowing the application to handle increased traffic and data loads by leveraging AWS services like Auto Scaling, Elastic Load Balancer, and Lambda. Reliability and Availability are maintained through AWS's global infrastructure and services such as Amazon S3, RDS Multi-AZ deployment, and Route 53, ensuring the website is accessible 24/7 with minimal downtime. Security is enforced at every level using HTTPS protocols, AWS Identity and Access Management (IAM), encryption in transit and at rest, and secure authentication with Amazon Cognito. Performance is optimized through content delivery using Amazon CloudFront, caching via Amazon ElastiCache, and efficient backend processing with Lambda. Maintainability is achieved by using microservices architecture, clean code practices, and modular design to allow easy updates and bug fixes. Usability ensures a smooth, responsive, and intuitive user interface for both desktop and mobile devices. Lastly, Cost-effectiveness is addressed by using AWS's pay-as-you-go model and serverless architecture to minimize infrastructure and maintenance expenses.

3.4 Feasibility Study

The development of a cloud-based eCommerce website on AWS is highly feasible from technical, economic, and operational perspectives. Technical feasibility is assured as AWS provides a wide range of reliable, scalable, and secure cloud services such as EC2, S3, RDS, Lambda, and Cognito, which support the complete deployment and operation of a modern eCommerce platform. These services reduce the complexity of infrastructure management and ensure high availability and performance. Economic feasibility is strong due to AWS's cost-efficient pay-as-you-go model, which eliminates the need for upfront hardware investments and reduces ongoing maintenance costs. This allows startups and small businesses to launch with minimal budget and scale as needed. Operational feasibility is achievable as the system is designed with user-friendly interfaces for both customers and administrators, with automated order processing, inventory management, and secure payment integrations. Additionally, the system can be easily maintained and updated with minimal downtime, supporting long-term sustainability. Overall, the project is practical and viable for deployment, offering a scalable and efficient solution for online retail businesses.

Economical Feasibility

The cloud-based eCommerce website is economically feasible due to the cost-effective nature of cloud computing and the pay-as-you-go pricing model offered by Amazon Web Services (AWS). Unlike traditional systems that require significant upfront investment in hardware, software licenses, and physical infrastructure, this solution allows businesses to pay only for the resources they use, minimizing initial costs. Operational costs are also reduced since AWS handles system maintenance, scalability, and security, eliminating the need for a large IT support team. Moreover, as the business grows, the system can be scaled easily without additional hardware investment, making it highly suitable for startups and small to medium enterprises. Overall, the project offers a budget-friendly approach with long-term financial sustainability and a high return on investment (ROI).

Technical Feasibility

The cloud-based eCommerce website is technically feasible due to the wide range of reliable and scalable services provided by Amazon Web Services (AWS). AWS offers all the necessary infrastructure and tools such as Amazon EC2 for virtual servers, Amazon S3 for file storage, Amazon RDS for database management, and AWS Lambda for running backend code without managing servers. These services are well-documented, widely adopted, and designed to work seamlessly together, ensuring the technical foundation of the project is strong. The use of AWS also enables built-in features like auto-scaling, load balancing, and real-time monitoring through CloudWatch, which are crucial for maintaining high performance and availability under varying workloads. Furthermore, the development team can use modern programming languages, frameworks, and DevOps practices supported by AWS, making the system easier to build, deploy, and maintain. Hence, from a technical standpoint, the project is fully achievable with current technologies and cloud resources.

Social Feasibility

The cloud-based eCommerce website is socially feasible as it aligns with the growing trend of online shopping and the increasing reliance on digital platforms for daily needs. It enhances user convenience by providing 24/7 access to products and services from anywhere, especially benefiting people in remote or busy areas. As more consumers embrace eCommerce, the platform is likely to be well-accepted and positively impact society by improving accessibility and customer experience.

4. SOFTWARE AND HARDWARE REQUIREMENTS

4.1 Software Requirements

To build and deploy a cloud-based eCommerce website on AWS, a combination of frontend, backend, database, and cloud platform software is needed.

- **Frontend**: HTML, CSS, JavaScript (React.js/Angular)
- **Backend**: Node.js / Python (Flask or Django)
- **Database**: Amazon RDS (MySQL/PostgreSQL) or DynamoDB
- **Storage**: Amazon S3 (for product images and files)
- **Authentication**: AWS Cognito
- **Serverless Functions**: AWS Lambda
- **API Management**: AWS API Gateway
- **Monitoring**: Amazon CloudWatch

4.2 Hardware Requirements

Since the eCommerce website is hosted on AWS cloud infrastructure, there is minimal need for physical hardware on the user or developer side. Users only require devices such as smartphones, tablets, or computers with internet access to interact with the website. On the development side, a standard development machine with internet connectivity is sufficient. All server-side hardware like servers, storage, and networking equipment is managed by AWS, providing scalable and reliable resources on demand.

- **User Devices**: Smartphones, tablets, laptops, desktops with internet access
- **Developer Machine**: PC or laptop with internet access and development tools installed
- **Cloud Infrastructure**: Managed by AWS (virtual servers, storage, networking)
- **Internet Connection**: Stable broadband for users and developers

5. SOFTWARE DESIGN

5.1 System Architecture

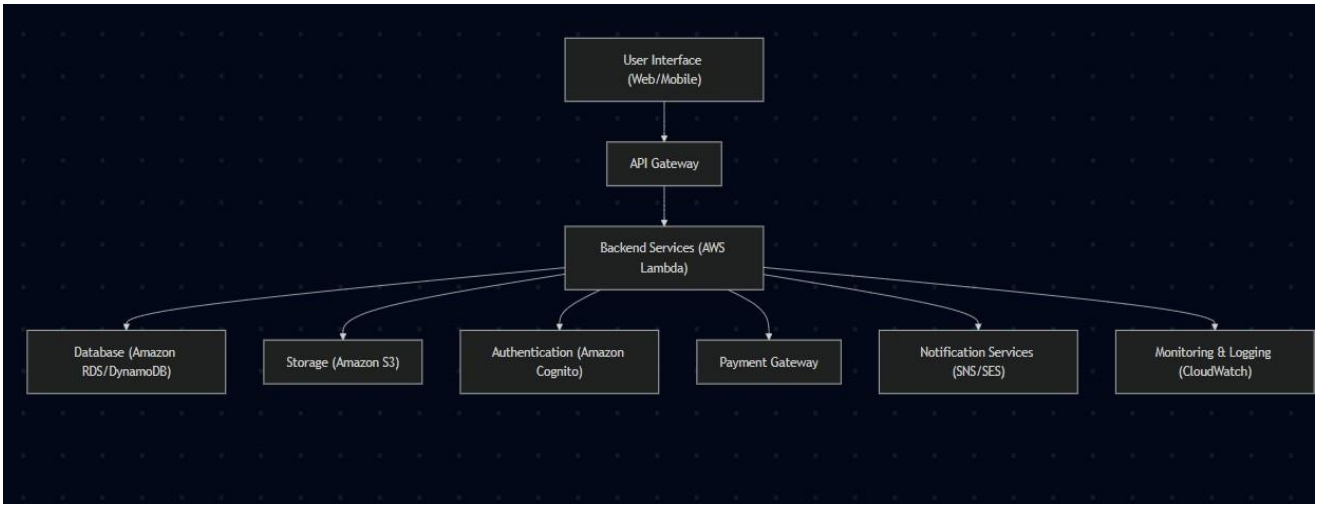

Figure:5.1 System Architecture

The system architecture of the cloud-based eCommerce website is designed using AWS services to ensure scalability, security, and high availability. Users interact through a web or mobile interface, which sends requests to the API Gateway. The API Gateway routes these requests to AWS Lambda functions that handle the backend logic, including user authentication (via Cognito), product management, and order processing. Data is securely stored in Amazon RDS or DynamoDB, while product images and static files reside in Amazon S3. Notifications are managed through SNS/SES, and the entire system's health and performance are monitored using CloudWatch.

5.2 Dataflow Diagram

In the eCommerce system, the user interacts with the frontend interface, which sends requests to the API Gateway. The API Gateway forwards these requests to backend AWS Lambda services that process authentication, database queries, file storage, payments, and notifications. Authentication data is verified using Amazon Cognito. Product and order information is stored and retrieved from the database, while images and files are saved in Amazon S3. Payment processing communicates with an external gateway, and notifications like order confirmations are sent to users via SNS or SES.

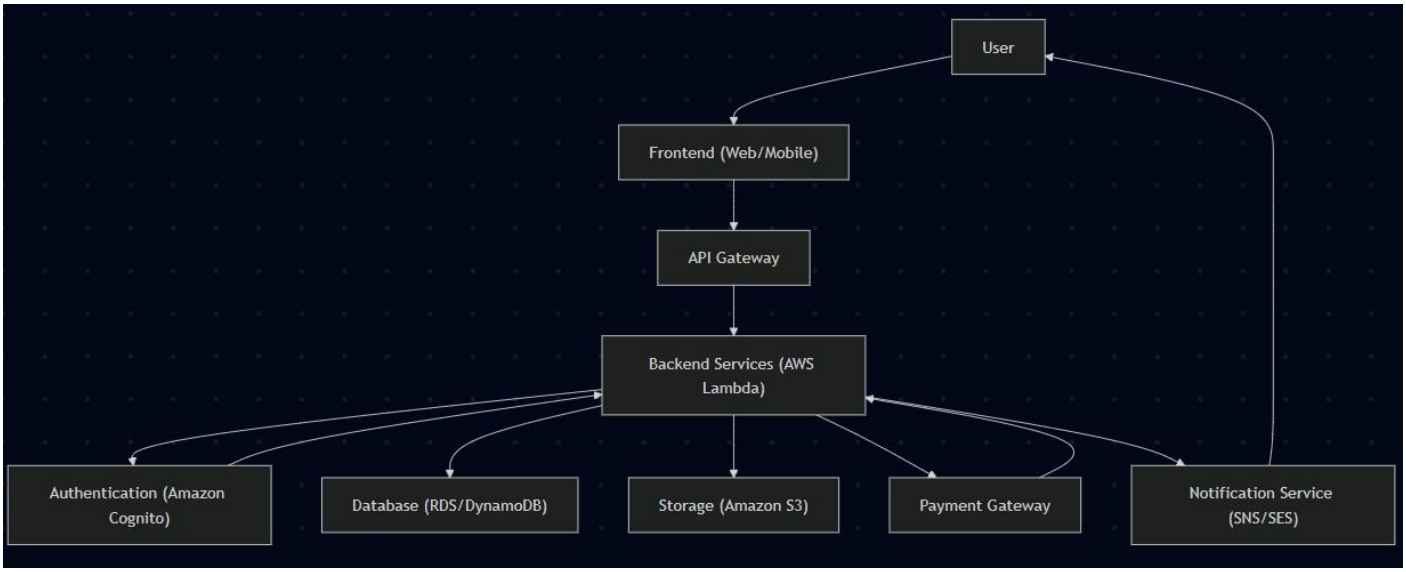

Figure:5.2 Dataflow Diagram

5.3 UML Diagrams

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

There are several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after (as part of documentation). UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. The two broadest categories that encompass all other types are:

- Behavioral UML diagram and
- Structural UML diagram.

As the name suggests, some UML diagrams try to analyses and depict the structure of a system or process, whereas other describe the behavior of the system, its actors, and its building components.

Goals: The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide a formal basis for understanding the modeling language.

Encourage the growth of OO tools market.

Integrate best practices.

The different types are as follows:

- Sequence diagram
- Use case Diagram
- Activity diagram
- Class diagram
- Collaboration diagram

Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

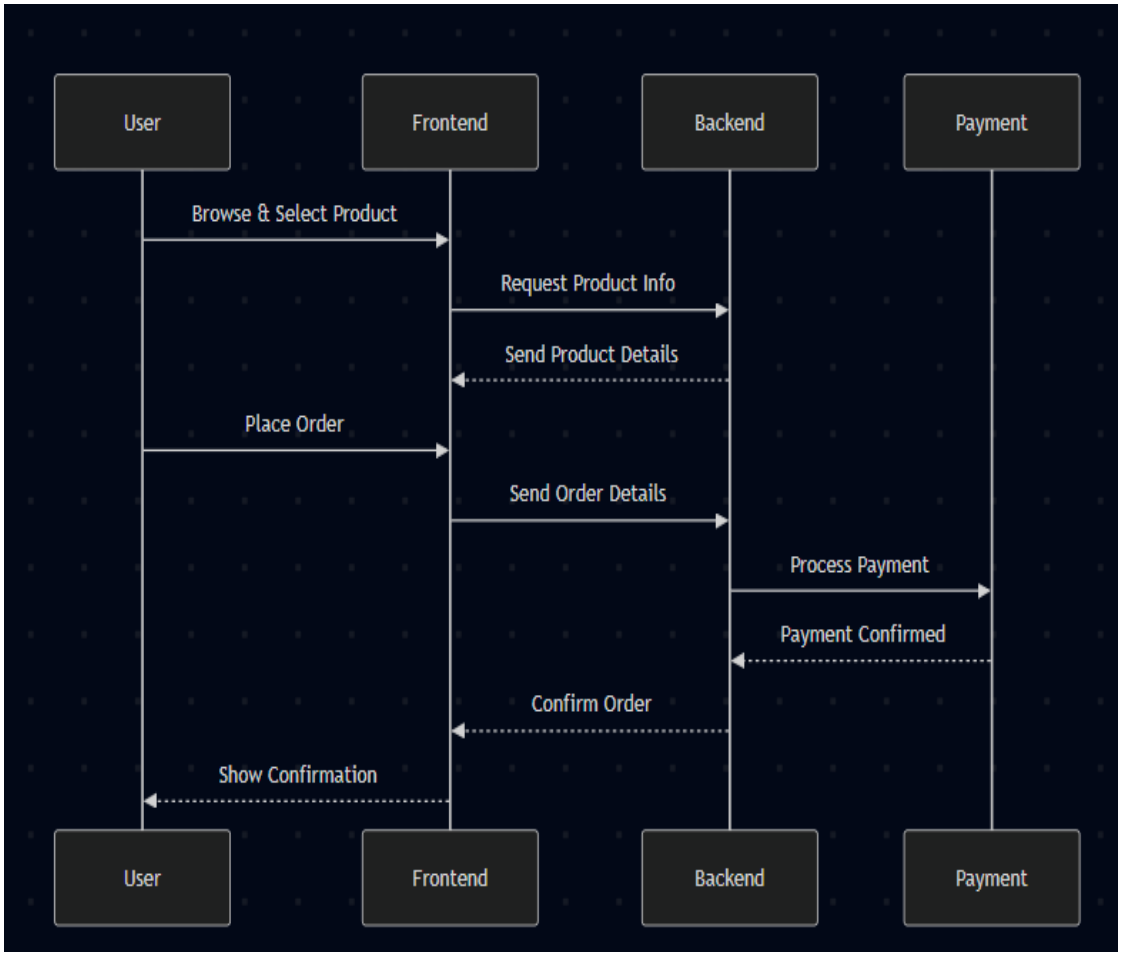

Figure 5.3.1 Sequence Diagram

Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use case in which the user is involved. A use case diagram is used to structure of the behavior thing in a model. The use cases are represented by either circles or ellipses.

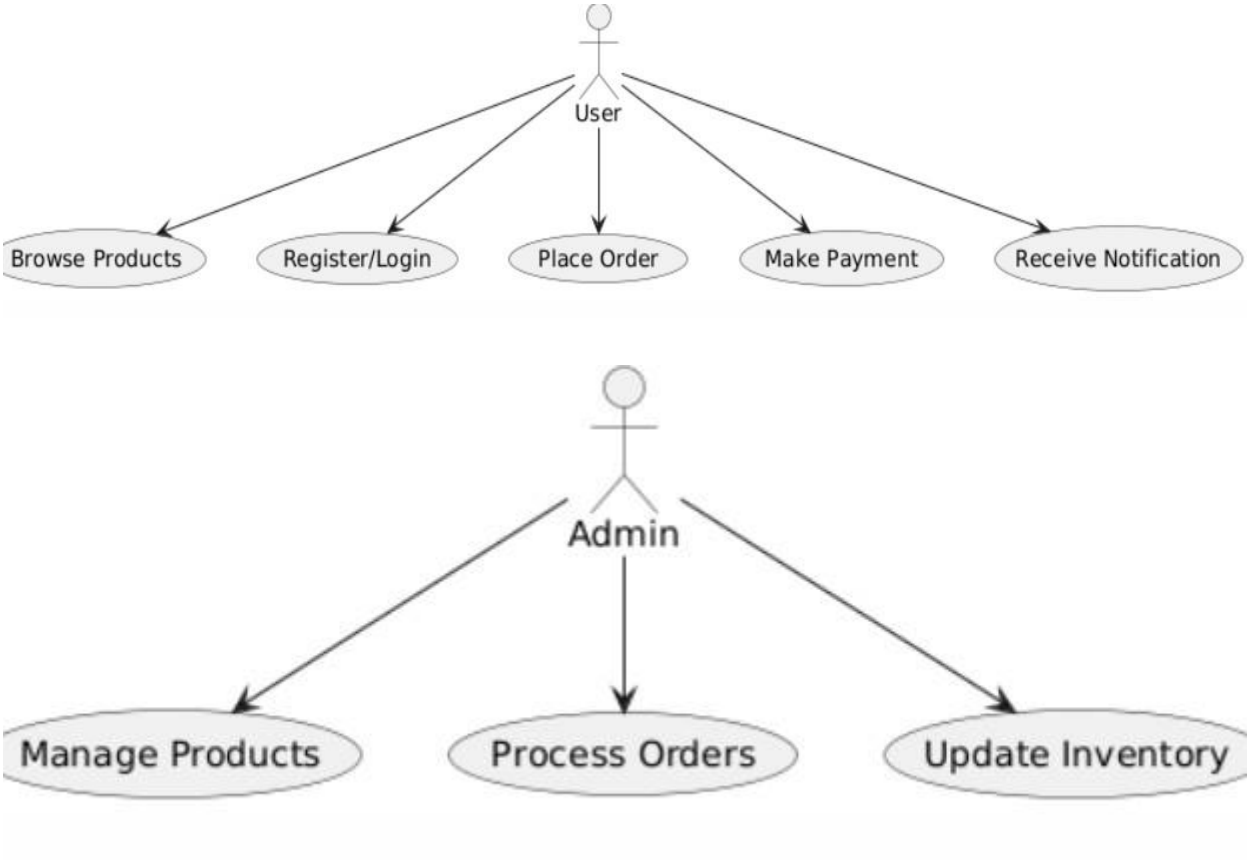

Figure 5.3.2 Use Case Diagram

Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

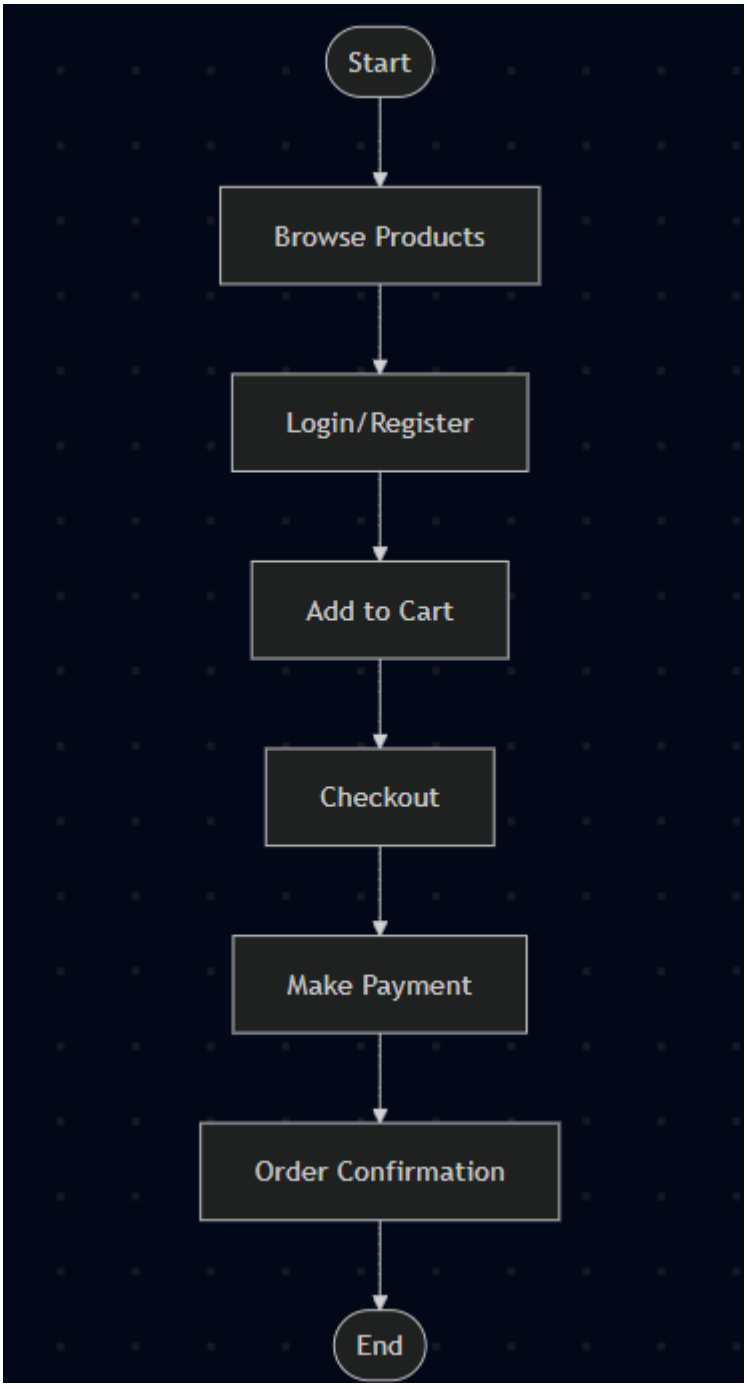

Figure 5.3.3 Activity Diagram

Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

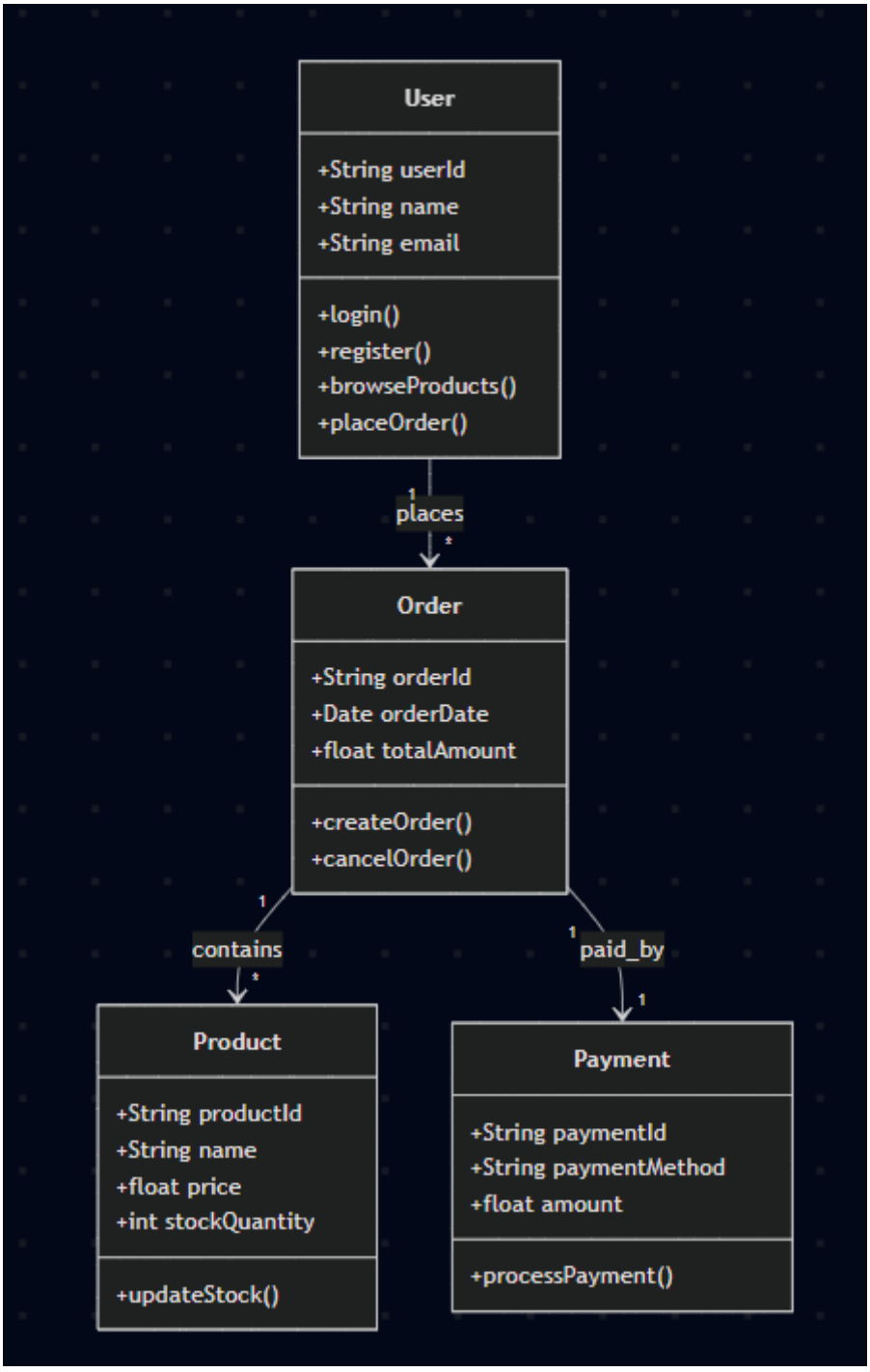

Figure 5.3.4 Class Diagram

6. CODING AND ITS IMPLEMENTATION

6.1 Source code

```
<!DOCTYPE html>
<html>
<head>
  <title>Cloud eCommerce</title>
</head>
<body>
  <h1>Welcome to Cloud eCommerce</h1>

  <h3>Register</h3>
  <input type="text" id="username" placeholder="Enter username">
  <button onclick="registerUser()">Register</button>

  <h3>Products</h3>
  <button onclick="fetchProducts()">Load Products</button>
  <ul id="productList"></ul>

  <script src="app.js"></script>
</body>
</html>
async function registerUser() {
  const username = document.getElementById("username").value;
  const res = await fetch("https://<API-GATEWAY-URL>/register", {
  method: "POST",
    body: JSON.stringify({ username }),
    headers: { "Content-Type": "application/json" },
  });
  const data = await res.json();
  alert(data.message);
}

async function fetchProducts() {
  const res = await fetch("https://<API-GATEWAY-URL>/products");
```

```
  const products = await res.json();
  const list = document.getElementById("productList");
  list.innerHTML = "";
  products.forEach(p => {
    const li = document.createElement("li");
    li.textContent = `${p.name} - $${p.price}`;
    list.appendChild(li);
  });
}
const AWS = require('aws-sdk');
const dynamo = new AWS.DynamoDB.DocumentClient();

exports.handler = async (event) => {
  const { username } = JSON.parse(event.body);

  await dynamo.put({
    TableName: "Users",
    Item: { username },
  }).promise();

  return {
    statusCode: 200,
    body: JSON.stringify({ message: "User registered successfully!" }),
  };
};
const AWS = require('aws-sdk');
const dynamo = new AWS.DynamoDB.DocumentClient();

exports.handler = async () => {
  const data = await dynamo.scan({ TableName: "Products" }).promise();
  return {
    statusCode: 200,
    body: JSON.stringify(data.Items),
  };
};
const AWS = require('aws-sdk');
```

```
const dynamo = new AWS.DynamoDB.DocumentClient();

exports.handler = async (event) => {
  const { username, productId } = JSON.parse(event.body);
  const orderId = Date.now().toString();

  await dynamo.put({
    TableName: "Orders",
    Item: {
      orderId,
      username,
      productId,
      orderDate: new Date().toISOString(),
    },
  }).promise();

  return {
    statusCode: 200,
    body: JSON.stringify({ message: "Order placed successfully!" }),
  };
};
```

6.2 Implementation

The implementation of the cloud-based eCommerce platform leverages Amazon Web Services (AWS) to provide a secure, scalable, and efficient infrastructure. The system is divided into three main layers: the presentation layer (frontend), the application logic layer (backend), and the data storage layer (database).

The frontend is built using HTML, CSS, and JavaScript, and is hosted on Amazon S3 as a static website. Amazon CloudFront is used as a content delivery network (CDN) to ensure faster and more reliable access to users across the globe by caching content closer to the user's location.

The backend logic is implemented using AWS Lambda, a serverless compute service that automatically manages the underlying infrastructure and scales automatically based on demand. Each Lambda function handles specific operations such as user registration, product listing, cart management, order processing, and payment handling. These functions are exposed via Amazon

API Gateway, which acts as the secure bridge between the frontend and backend, handling HTTP requests and routing them to the appropriate Lambda functions.

For data storage, Amazon DynamoDB is used as the primary NoSQL database to store user data, product information, and order history. DynamoDB offers fast and predictable performance with seamless scalability. The use of Amazon Cognito provides secure user authentication and authorization, enabling features like user sign-up, login, and session management without building a custom authentication mechanism.

To process payments, the system integrates with third-party payment gateways (like Razorpay, Stripe, or PayPal), which are called securely from the backend Lambda functions. AWS Simple Notification Service (SNS) or Amazon SES (Simple Email Service) is used to send order confirmation alerts or emails to users.

Additionally, AWS CloudWatch is used for monitoring logs and application performance, ensuring real-time alerts and operational insights. The entire infrastructure can be managed using AWS CloudFormation or Terraform, allowing for infrastructure as code (IaC), which simplifies deployment and version control.

This implementation ensures:

- High availability and scalability
- Cost-effectiveness due to the serverless model
- Improved security with Cognito and IAM roles
- Fast performance with globally distributed CloudFront and DynamoDB

7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

7.1. Types Of Tests

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application

Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Inputs: contain all required fields (company Name, role, required Skills, and resume) with different scenarios to cover cases like missing company name, missing role, missing resume, and various skill checks.

Output: provide meaningful feedback according to the input scenario: confirmation of suitability or identification of missing fields or skills.

Functions:

checkResume(),document.getElementById(),split(), trim(), toLowerCase().

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.
Procedures	: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.
The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.
Test Results**:** All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.
Test Results**:** All the test cases mentioned above passed successfully. No defects encountered.

7.2 Test Cases:

Test Case ID	Test Case Description	Input	Expected Output	Status
TC01	User Registration	Username, Email, Password	"Registration Successful" message	Pass/Fail
TC02	User Login	Email, Password	Redirect to homepage/dashboard	Pass/Fail
TC03	Browse Products	-	List of available products displayed	Pass/Fail
TC04	Add Product to Cart	Selected product ID	Product added to cart confirmation	Pass/Fail
TC05	Place Order	Product ID, User Info	Order placed and confirmation received	Pass/Fail
TC06	Payment Gateway Integration	Card/UPI details	Payment successful message	Pass/Fail
TC07	Admin Login	Admin credentials	Redirect to admin dashboard	Pass/Fail
TC08	Product Addition by Admin	Product details	Product added to database	Pass/Fail
TC09	View Order History	User login	List of previous orders	Pass/Fail
TC10	Search for Products	Search keyword	Matching products displayed	Pass/Fail

Table no 7.2 Test Cases

Test Case 1:

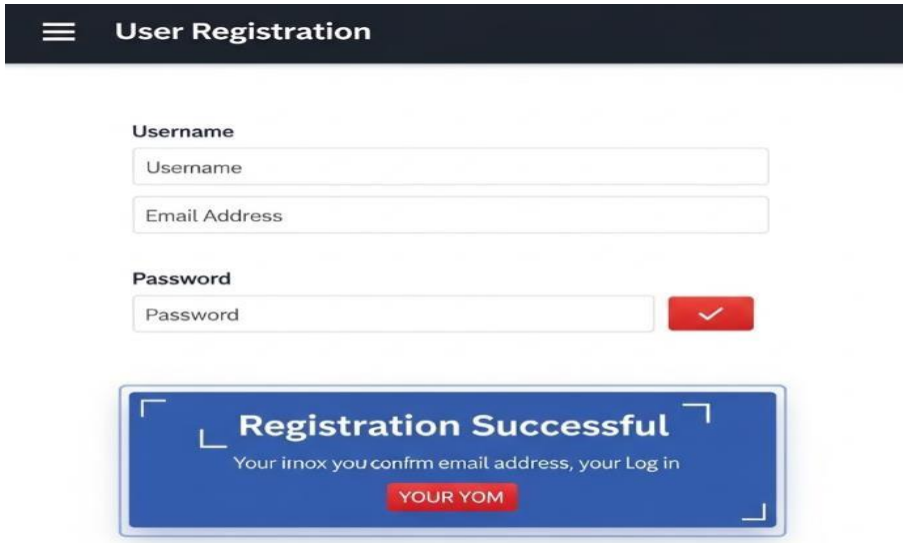

Figure 7.2.2: Test Case 1

Test Case 2:

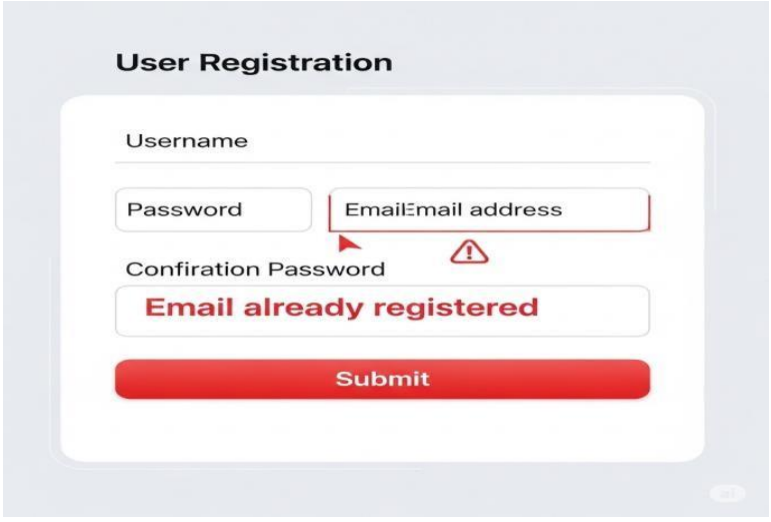

Figure 7.2.3: Test Case 2

Test Case 3:

Figure 7.2.4: Test Case 3

Test Case 4:

Figure 7.2.5: Test Case 4

Test Case 5:

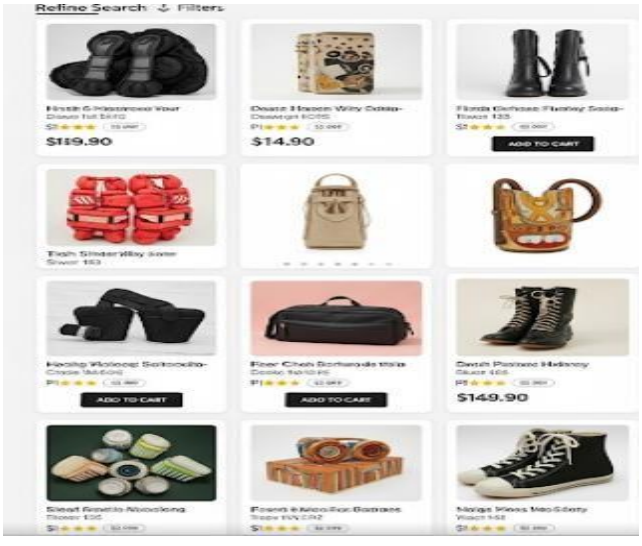

Figure 7.2.9: Test Case 8

Test Case 6:

Figure 7.2.7: Test Case 6

Test Case 7:

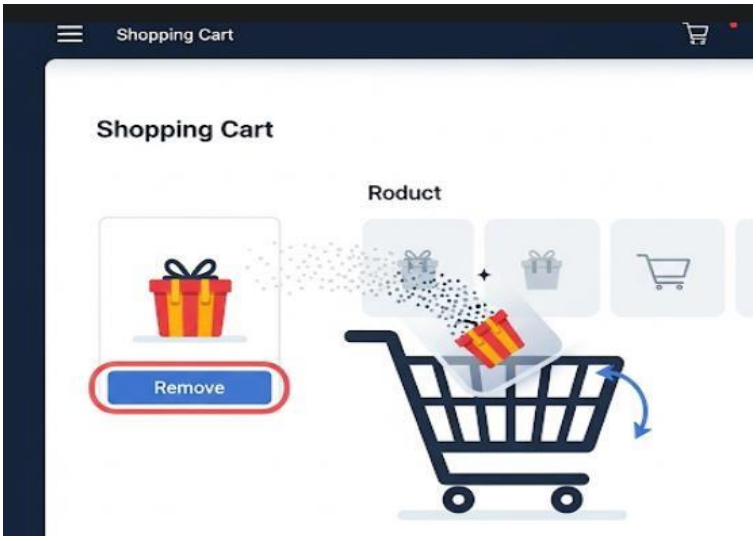

Figure 7.2.8: Test Case 7

Test Case 8:

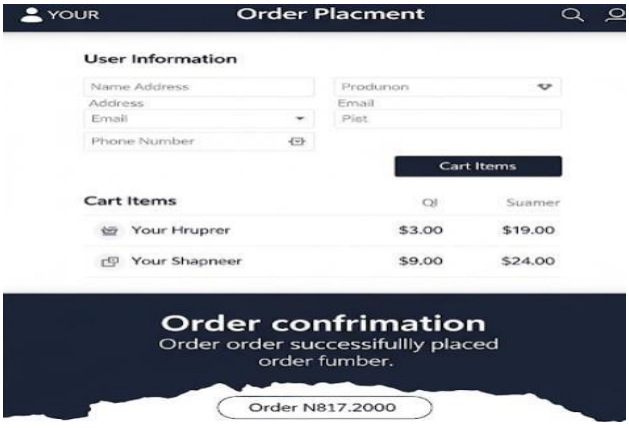

Figure 7.2.9: Test Case 8

Test Case 9:

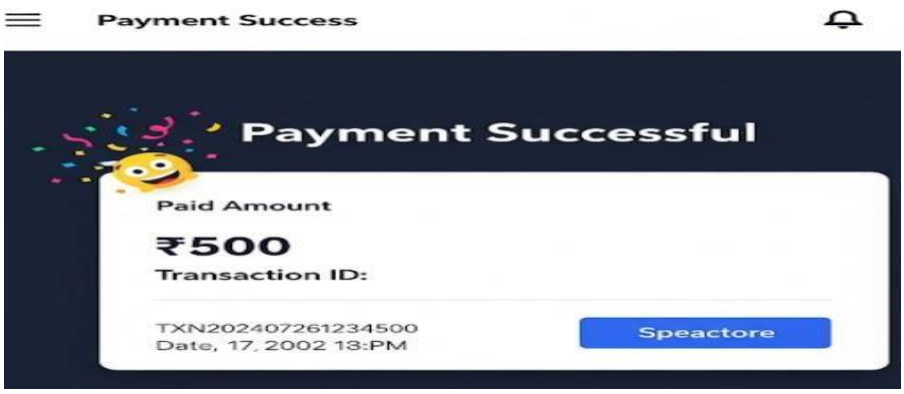

Figure 7.2.10: Test Case 9

Test Case 10:

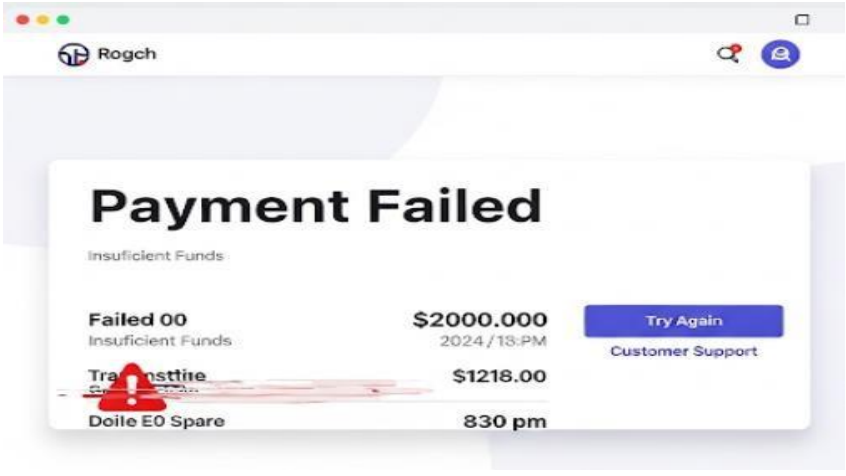

Figure 7.2.11: Test Case 10

Test Case 11:

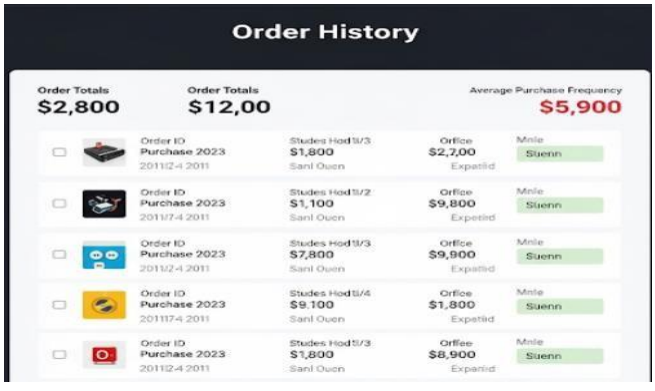

Figure 7.2.12: Test Case 11

8. OUTPUT SCREENS

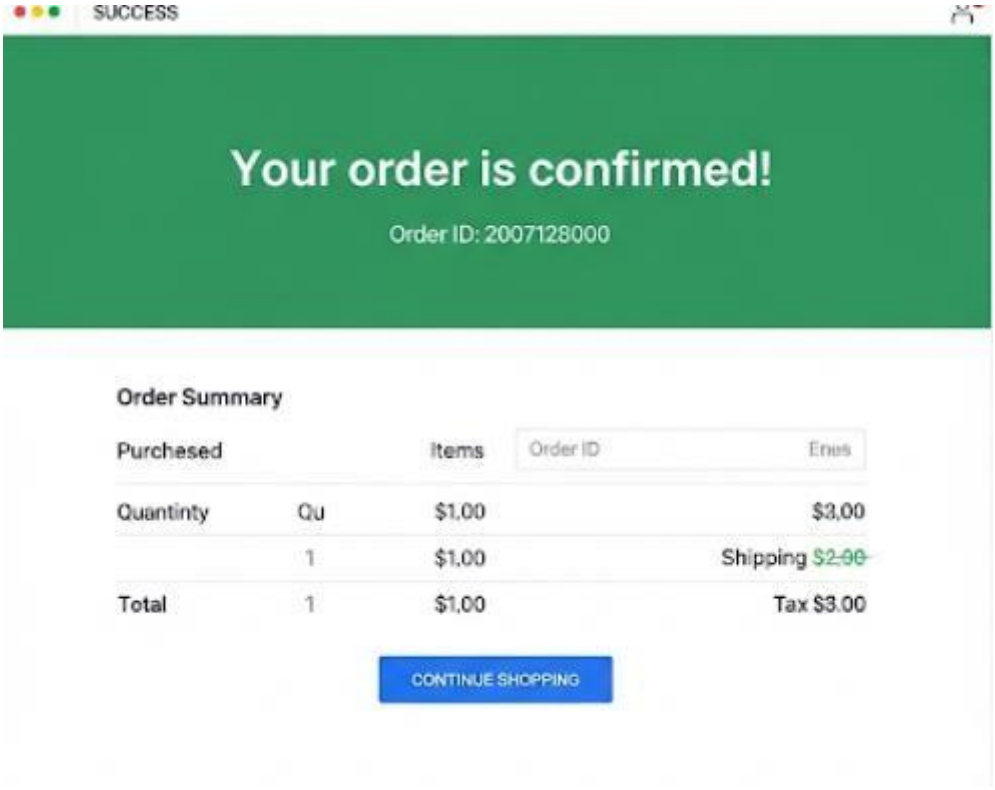

Figure 8.1: Out Screen-1

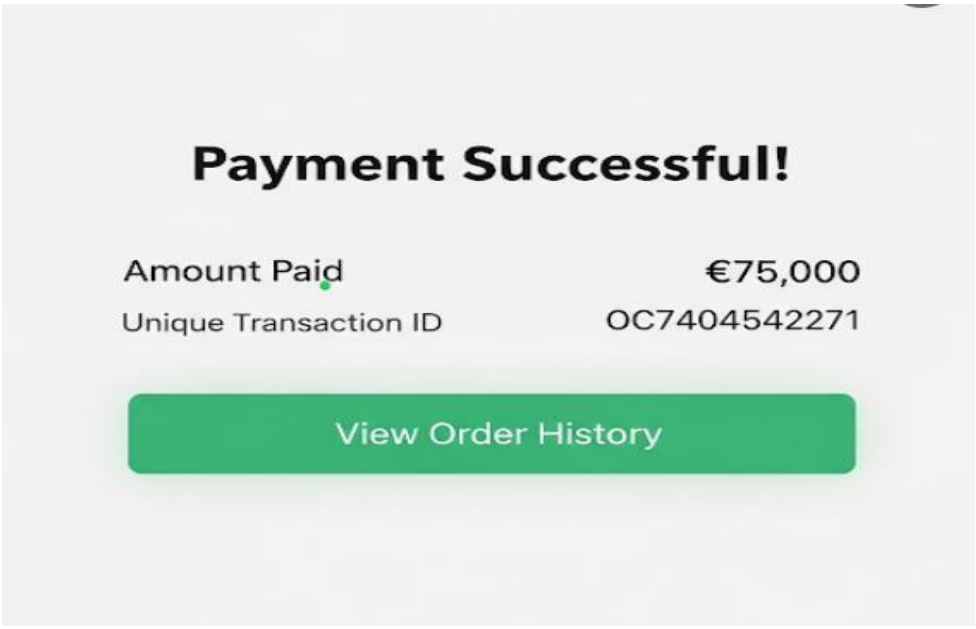

Figure 8.2: Output screen-2

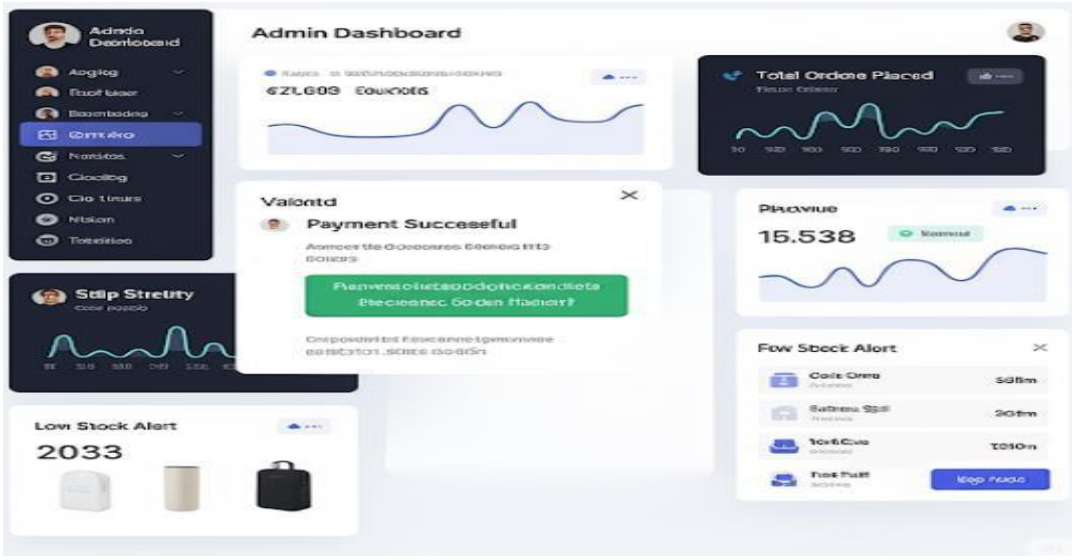

Figure 8.3: Output screen-3

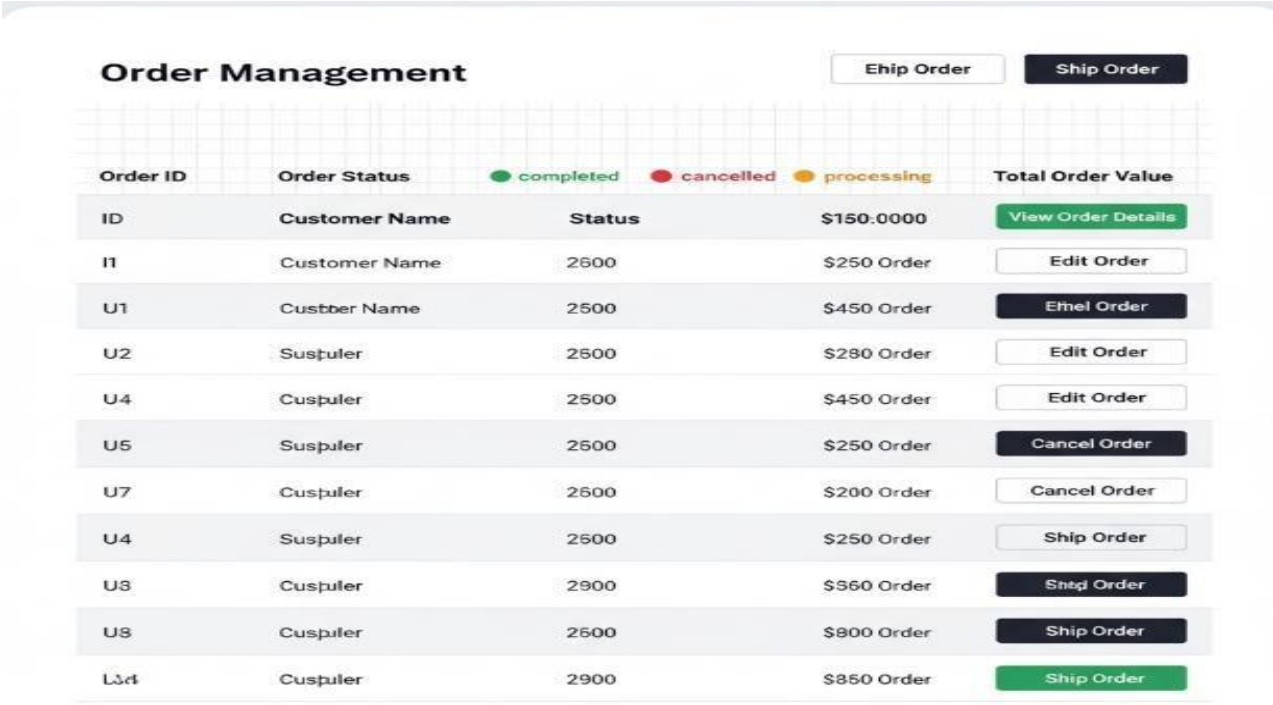

Figure 8.4: Output screen-4

9. CONCLUSION

In conclusion, this cloud-based e-commerce website project demonstrates the effective integration of modern web technologies with AWS cloud infrastructure to deliver a fully functional, scalable, and user-friendly online shopping experience. By utilizing core AWS services such as Amazon S3 for frontend hosting, AWS Lambda and API Gateway for serverless backend operations, and Amazon RDS or DynamoDB for secure and fast data management, the platform ensures high availability, low latency, and seamless interaction across modules. The application focuses on enhancing user satisfaction by implementing intuitive screens like Order Confirmation and Payment Successful pages, which reassure users about the completion of their transactions and enhance trust in the platform.

On the administrative side, the inclusion of a feature-rich Admin Dashboard and Order Management System enables backend users to monitor orders, track performance metrics, and make informed decisions in real-time. These interfaces were thoughtfully designed to prioritize clarity, ease of navigation, and quick access to critical actions such as viewing, updating, or processing orders. The system also supports modular enhancements, making it adaptable to additional features like inventory alerts, customer feedback, and personalized recommendations.

Moreover, the project reflects the broader advantages of deploying an e-commerce application on the cloud — such as automatic scaling, cost-efficiency, easy integration of third-party APIs (e.g., payment gateways), data durability, and built-in security via AWS Identity and Access Management (IAM). The architecture not only supports present-day functionality but also lays the groundwork for advanced future integrations, such as AI-based product recommendations, real-time analytics, and multilingual/multi-currency support.

Overall, this project provides a strong foundation for a robust e-commerce platform that balances technical reliability, business logic, and user-centered design. It stands as an excellent example of how cloud technologies can transform traditional web development into dynamic, real-time, and cost-effective digital ecosystems.

.

10. FUTURE ENHANCEMENTS

Looking ahead, the e-commerce website built on AWS offers several opportunities for further enhancement and growth. One major improvement would be the integration of AI-based product recommendation systems to personalize the shopping experience based on user behavior, past purchases, and browsing patterns. This can significantly boost user engagement and sales conversion rates.

To make the platform more inclusive and globally accessible, support for multiple languages and currencies could be added. This would enable users from different regions to navigate and transact in their preferred language and currency, improving the overall usability and reach of the application.

Another potential enhancement is the development of dedicated mobile applications for Android and iOS, which would offer a more optimized and accessible shopping experience. These apps could be seamlessly integrated with the existing backend through APIs or AWS Amplify.

Implementing a real-time analytics dashboard for the admin using services like Amazon QuickSight can provide valuable insights into user activity, sales trends, and inventory levels. This data-driven approach would support better business decisions and performance monitoring.

Security can also be further strengthened by incorporating multi-factor authentication (MFA), CAPTCHA protection, and enhanced access controls using AWS Cognito or IAM policies.

Inventory management can be improved by automating low-stock alerts and integrating with supplier systems to streamline restocking processes. Additionally, the introduction of a loyalty program and promotional engine would allow the platform to reward repeat customers and offer targeted discounts, thereby increasing customer retention.

To improve customer service, integrating a chatbot or live chat support system could provide users with immediate assistance for product inquiries or order issues. Furthermore, shipment tracking integration would allow users to view real-time updates on their orders directly through the website.

Lastly, automating periodic reports and email notifications using AWS Lambda and Simple Email Service (SES) would ensure that administrators are consistently informed about key metrics and business updates. These enhancements would not only improve the technical robustness of the system but also contribute to a more professional and satisfying user and administrative experience.

11. REFERENCES

[1]. “Indian e-commerce market to grow by 21.5% in 2022, forecasts Global Data”, jan 21,2022, https://www.globaldata.com/indian-e-commerce-market-grow-21-5-2022-forecasts-globaldata/

[2]. “Types of E-Commerce”, https://www.indiafilings.com/learn/types-of-e-commerce/

[3]. ”Amazon hosting services” https://aws.amazon.com/what-is/web-hosting/

[4]. Le Shen, Research on E-commerce Website Design Based on User Experience——Taking Online Digital Printing as an Example[D], East China University of Science and Technology, 2013

[5]. Lu Du, Xia Zhong, Research on UGC type models of socialized e-commerce websites based on user experience[J], Design, 2014(4), 82-84

[6]. Chaudhury, A.,(2002),“e-Business and E-Commerce Infrastructure Technologies Support the e- Business Initiative”

[7]. Gupta, A. (2014, January). E-Commerce : Role Of ECommerce In Today's Business. International Journal of Computing and Corporate Research, 4(1)

[8]. “Indian e-commerce market to grow by 21.5% in 2022, forecasts Global Data”, jan 21,2022, https://www.globaldata.com/indian-e-commerce-market-grow-21-5-2022-forecasts-globaldata/

[9]. ”Amazon hosting services” https://aws.amazon.com/what-is/web-hosting/

[10] Anitha Eemani,(2019). Network Optimization and Evolution to Bigdata Analytics Techniques, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 8, Issue 1, January 2019

[11] Ketulkumar Govindbhai Chaudhari. (2019). Review on Challenges and Advanced Research Areas in Internet of Things. International Journal of Innovative Research in Computer and Communication Engineering, 7(7), 3570-3574. DOI: 10.15680/IJIRCCE.2019. 0707016.

[12] Yeshwanth Valaboju (2017). A Review on the Database Security Requirements and Guidelines, International Journal of Scientific Research in Science and Technology, Volume 3, Issue 6, July-August 2017

[13] Soni, Vishal Dineshkumar,(2018) Role of AI in Industry in Emergency Services. International Engineering Journal For Research & Development, 3(2), 6. https://doi.org/10.17605/OSF.IO/C67BM

[14] Ankit Narendrakumar Soni (2018). Smart Devices Using Internet of Things for Health Monitoring. International Journal of Innovative Research in Science, Engineering and Technology, 7(5), 6355-6361. doi:10.15680/IJIRSET.2018.0705233.

[15] BhagyaRekha Kalukurthi, "A Comprehensive Review on Machine Learning and Deep Learning", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 8, Issue 6, June 2019

[16] Jubin Dipakkumar Kothari (2018). Garbage Level Monitoring Device Using Internet of Things with ESP8266, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 7, Issue 6, June 2018, 2995-2998.

[17] Ketulkumar Govindbhai Chaudhari. (2019). Water Quality Monitoring System using Internet of Things and SWQM Framework. International Journal of Innovative Research in Computer and Communication Engineering, 7(9), 3898-3903. DOI: 10.15680/IJIRCCE.2019. 0709008

[18] Vishal Dineshkumar Soni. (2019). IOT connected with e-learning. International Journal on Integrated Education, 2(5), 273-277. https://doi.org/10.31149/ijie.v2i5.496

[19] Rakesh Rojanala,(2017). Machine Learning: Intersection of Statistics and Computer Science. International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 8, August 2017.

[20] Soni, Ankit Narendrakumar, Diabetes Mellitus Prediction Using Ensemble Machine Learning Techniques (July 3, 2020). Available at SSRN: https://ssrn.com/abstract=3642877 or http://dx.doi.org/10.2139/ssrn.3642877.