Cloud-Based E-Commerce Platform

1	NATNAEL ENDALE	ETS1008/13
2	KIRUBEL ATEKA	ETS0734/13
3	KIRUBEL DAGNACHEW	ETS0735/13
4	NATNAELMULUGETA	ETS1020/13
5	DESALEGN SENDEK	ETS1553/13
6	MITIKU ABEBE	ETS0904/13
7	LAELAY TEMESGEN	ETS0746/13

AASTU, Department of Software Engineering
Introduction to Cloud Computing

Submission date May 8,2025

Acknowledgement

We would like to express our sincere gratitude to our instructor and peers for their guidance and support throughout the development of this cloud-based e-commerce platform. Special thanks to the Render team for providing an efficient cloud deployment solution.

Table of Contents

Cloud-Based E-Commerce Platform 1

Acknowledgement 2

Chapter One: Introduction 5

1.1 Purpose of this Document 5

1.2 Beneficiaries of the Product 5

1.3 Target Audience of This Document 5

Chapter Two: Project Overview 5

2.1 Problem Statement 5

2.2 Motivation 5

2.3 Objectives 5

2.3.1 General Objective 5

2.3.2 Specific Objectives 6

2.4 Scope 6

2.4.1 In Scope 6

2.4.2 Out of Scope 6

2.5 Key Features and Benefits 6

Chapter Three: System Architecture 6

3.1 High-Level Overview 6

3.2 Component Descriptions 6

3.2.1 Frontend (Next.js with Tailwind CSS) 6

3.2.2 Backend (Next.js API Routes) 7

3.2.3 Authentication (OAuth) 7

3.2.4 Database (MongoDB Atlas) 7

3.3 Data Flow and Integration Points 7

Chapter Four: Cloud Infrastructure 7

4.1 Chosen Cloud Platform and Services 7

4.2 Justification for Each Service 7

4.3 Infrastructure-as-Code or Deployment Scripts 7

Chapter Five: API Reference 8
5.1 Authentication & Authorization 8
5.2 User Management 8
5.3 Product and Cart Services 8
Chapter Six: Frontend Guide 8
6.1 Installation and Setup 8
6.2 UI/UX Overview and Design 8
Chapter Seven: Backend Guide 8
7.1 Installation and Setup 8
7.2 Database Schema and Migrations 9
7.3 Business Logic Modules and Services 9
7.3.1 Controllers 9
7.3.2 Services 9
Chapter Eight: Testing & Quality Assurance 9
8.1 Testing Strategy 9
8.2 Tools and Frameworks 9
Chapter Nine: Deployment & Operations 9
9.1 Deployment Process 9
9.2 Monitoring and Logging Setup 9
9.3 Security Best Practices and Compliance 10
Chapter Ten: Future Work & Extensions 10
10.1 Planned Enhancements 10
10.2 Potential Integrations 10

Chapter One: Introduction

1.1 Purpose of this Document

This document provides a detailed overview of a cloud-based e-commerce platform built using modern web technologies and hosted on the Render cloud platform. It outlines the architecture, features, cloud infrastructure, and development process.

1.2 Beneficiaries of the Product

- Small and medium-sized businesses wanting to move online.
- Developers looking for scalable e-commerce boilerplate solutions.
- Students and educators using cloud-based projects for learning.

1.3 Target Audience of This Document

- Software developers
- Cloud computing students
- Technical project stakeholders
- Academic instructors

Chapter Two: Project Overview

2.1 Problem Statement

Setting up an online store is costly and technically challenging for many small businesses. Traditional hosting services also lack scalability and uptime.

2.2 Motivation

With the rise of cloud platforms, it is possible to create affordable and scalable e-commerce systems with minimal DevOps overhead.

2.3 Objectives

2.3.1 General Objective

Develop a fully functional cloud-based e-commerce platform using Render and other modern technologies.

2.3.2 Specific Objectives

- Use Tailwind CSS and Next.js for the frontend.
- Use MongoDB Atlas for flexible cloud-based data storage.
- Deploy using Render with CI/CD support.
- Implement secure OAuth-based authentication.

2.4 Scope

2.4.1 In Scope

- User registration and login
- Product listings and cart
- Cloud-based hosting and authentication

2.4.2 Out of Scope

- Payment gateway integration
- Mobile application
- Admin dashboard (planned for future)

2.5 Key Features and Benefits

- Cloud-hosted for high availability
- OAuth login for improved security
- Serverless backend via Next.js API routes
- Responsive UI with Tailwind

Chapter Three: System Architecture

3.1 High-Level Overview

The system follows a serverless web architecture with Next.js powering both frontend and backend logic, MongoDB Atlas for storage, and Render for deployment.

3.2 Component Descriptions

3.2.1 Frontend (Next.js with Tailwind CSS)

Responsive UI with pages for products, login, cart, and checkout.

3.2.2 Backend (Next.js API Routes)

Handles product CRUD operations, user sessions, and cart logic.

3.2.3 Authentication (OAuth)

Google OAuth is used for secure sign-in and session handling.

3.2.4 Database (MongoDB Atlas)

Document-oriented NoSQL database hosted in the cloud.

3.3 Data Flow and Integration Points

- Frontend communicates with API routes via fetch.
- API interacts with MongoDB Atlas and OAuth services.
- Render deploys both frontend and backend under one domain.

Chapter Four: Cloud Infrastructure

4.1 Chosen Cloud Platform and Services

- **Render**: Hosting both frontend and backend.
- **MongoDB Atlas**: Cloud database.
- **OAuth (Google)**: Authentication and identity provider.

4.2 Justification for Each Service

- **Render** was chosen for its simplicity and free-tier support, suitable for fast deployments without complex setup like AWS EC2 or Azure VMs.
- **MongoDB Atlas** offers a free tier with cloud backup, security, and excellent integration with JavaScript/Node.js projects.
- **OAuth (Google)** allows quick, secure sign-in without handling passwords.

4.3 Infrastructure-as-Code or Deployment Scripts

- Deployment done using GitHub auto-deploy integration with Render.
- Environment variables managed through Render’s dashboard.

Chapter Five: API Reference

5.1 Authentication & Authorization

- `GET /api/auth/login`: Initiate Google login
- `GET /api/auth/callback`: OAuth redirect handler

5.2 User Management

- `GET /api/user`: Retrieve current user session

5.3 Product and Cart Services

- `GET /api/products`: Get all products
- `POST /api/cart`: Add to cart
- `DELETE /api/cart/:id`: Remove item

Chapter Six: Frontend Guide

6.1 Installation and Setup

```
git clone <repo-url>
npm install
npm run dev
```

6.2 UI/UX Overview and Design

- Homepage with product cards
- Login button with Google
- Shopping cart on the navbar

Chapter Seven: Backend Guide

7.1 Installation and Setup

Same as frontend (Next.js is fullstack).

7.2 Database Schema and Migrations

- Products: `{ name, price, description, imageURL }`
- Cart: `{ userId, productId, quantity }`

7.3 Business Logic Modules and Services

7.3.1 Controllers

- Handle API requests (e.g., `/api/products`, `/api/cart`)

7.3.2 Services

- MongoDB client connections
- Session verification

Chapter Eight: Testing & Quality Assurance

8.1 Testing Strategy

- Manual testing via Postman and browser
- Automated testing (Planned using Jest)

8.2 Tools and Frameworks

- Postman, Vercel DevTools, Browser DevTools
-

Chapter Nine: Deployment & Operations

9.1 Deployment Process

- Code pushed to GitHub
- Auto-deployed to Render

9.2 Monitoring and Logging Setup

- Render logs available in dashboard
- MongoDB Atlas activity logs

9.3 Security Best Practices and Compliance

- HTTPS by default
- OAuth for login
- No passwords stored
- Environment variables hidden

Chapter Ten: Future Work & Extensions

10.1 Planned Enhancements

- Admin dashboard
- Inventory tracking
- Payment integration

10.2 Potential Integrations

- Stripe/PayPal for payments
- WhatsApp for order updates
- Mobile app (Flutter/React Native)