TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my deepest appreciation to all those who played a vital role in the successful completion of my project, **Flutter-Based E-Commerce Web Application with Firebase**. This project provided valuable experience in web development using Flutter, Firebase Authentication, and real-time hosting, and I am sincerely grateful for the constant support and encouragement throughout this journey.

I extend my heartfelt thanks to **Mr. Krishan Tulli**, Head of Department, **MCA**, for his invaluable guidance and consistent support, which greatly influenced the successful outcome of my work. I am also thankful to all the faculty members of the **UIC department** for their thoughtful suggestions and encouragement, which enriched my learning experience.

I am especially grateful to my project supervisor, **Mr. Sachin Raj**, Assistant Professor at the **University Institute of Computing**, for his expert mentorship and continuous motivation. His readiness to provide feedback and resolve doubts helped me navigate technical challenges and strengthen the project's foundation.

A special thanks to my **parents**, whose unwavering support, belief, and encouragement kept me motivated throughout the development process.

To everyone who directly or indirectly contributed to the success of the **Flutter-Based E-Commerce Web Application**, your guidance and support have been truly invaluable. I feel fortunate to have received such meaningful mentorship and support from all corners.

Sachin Raj

UID: 22MCA20641

ABSTRACT

This project presents **an innovative Flutter-based e-commerce web application** designed to deliver a seamless online shopping experience while integrating modern technologies like **Firebase Authentication** and **SQL database support**. The application serves as a user-friendly platform that bridges the gap between buyers and products through an intuitive and responsive interface developed using the Flutter framework.

At its core, the application features a secure **login and signup system powered by Firebase Authentication**, ensuring that users can register and access their accounts with ease. Additionally, real-time data interactions and form submissions are handled via **Firebase and SQL APIs**, enabling dynamic features such as product detail management, cart functionalities, wishlist tracking, and secure checkouts.

The platform includes modules such as product browsing, detailed product views, cart management, checkout processing, and order history, all designed to simulate a full-fledged e-commerce experience. The use of **Firebase Analytics** allows administrators to monitor user engagement, while SQL-based backend integration supports dynamic data handling, storage, and retrieval.

Designed with scalability in mind, this web application provides a clean user interface, fast load times, and modular architecture, making it adaptable to various business models and easy to extend with future updates such as vendor dashboards, promotional campaigns, or payment gateway integrations.

By combining the power of Flutter, Firebase, and SQL, this project demonstrates a comprehensive and practical approach to building modern e-commerce platforms that are efficient, secure, and user-centric.

CHAPTER 1.
INTRODUCTION

In today's digital era, e-commerce has become an essential component of global trade, revolutionizing the way businesses interact with consumers. With increasing demand for online shopping and the convenience it offers, the need for a reliable, scalable, and user-friendly e-commerce platform is more vital than ever. This project, titled **"Flutter-Based E-Commerce Web Application with Firebase and SQL Integration,"** aims to provide a comprehensive solution for both businesses and consumers by developing a modern web-based shopping platform that combines intuitive design with powerful backend capabilities.

Built using the Flutter framework, this web application ensures a smooth and responsive user experience across multiple screen sizes and devices. Flutter's cross-platform capabilities allow for consistency in UI and performance while reducing development time. The integration of **Firebase Authentication** ensures secure and seamless login and signup processes for users, eliminating the risks associated with unauthorized access and enhancing data protection.

In addition to authentication, this project utilizes **Firebase Analytics** to monitor user behavior and engagement on the platform, helping administrators gather valuable insights into user preferences, purchase patterns, and feature performance. The integration of **SQL-based backend** ensures efficient management of structured data such as product listings, user profiles, order histories, and inventory management, enabling reliable CRUD operations and scalability for future growth.

1.1 Identification of Problem

In today's fast-paced digital economy, the demand for intuitive, secure, and feature-rich e-commerce platforms is ever-growing. However, despite the vast number of online retail platforms available, several challenges continue to affect both businesses and consumers, particularly for emerging or medium-scale vendors.

Our **Flutter-Based E-Commerce Web Application with Firebase and SQL Integration** addresses this gap by tackling key problems encountered in modern online shopping systems.

1. **Fragmented User Experience:** Many e-commerce platforms lack seamless integration across web and mobile interfaces. Users expect responsive, intuitive design that adapts across devices, which is often lacking in traditional systems.

2. **Authentication and Data Security:** Ensuring secure user authentication and safe handling of user data is a critical challenge. Many platforms struggle with implementing robust authentication mechanisms.
3. **Absence of Real-Time Features:** Real-time cart management, product availability updates, and instant order confirmation are essential features in today's e-commerce world, which are often limited in conventional systems.
4. **Lack of Insightful Analytics:** Tracking user behavior, sales data, and traffic flow is crucial for strategic decisions. However, many systems either lack analytics or fail to present meaningful insights.
5. **Hosting and Deployment Limitations:** Traditional hosting environments can limit scalability and may not support fast deployment pipelines. Many small businesses require quick-to-deploy solutions with minimal maintenance.

1.2 Approach to solve the Problem

To effectively address these challenges, our approach focuses on modern development practices using **Flutter**, **Firebase**, and **SQL**, creating a unified and scalable e-commerce platform with secure, real-time capabilities and enhanced user experience.

1. **Cross-Platform Flutter UI:** By using Flutter, we ensure responsive design and performance across web and mobile. The UI is clean, interactive, and highly customizable, ensuring a pleasant experience for users browsing or shopping from any device.
2. **Firebase Authentication Integration:** Firebase Authentication secures the login and signup process using Google's trusted platform. Users can register and log in using their email or social accounts, ensuring secure and streamlined access to the platform.
3. **Real-Time Firebase Features:** Integration with Firebase provides real-time sync for user activities like adding items to cart, order confirmation, and user notifications. This enhances engagement and ensures up-to-date data without reloading.
4. **Netlify Deployment for Hosting:** The project is hosted on Netlify for fast, free, and secure deployment. This ensures quick public access, automated CI/CD pipeline, and scalable hosting without backend server management.

1.3 Timeline

- **Phase 1: Planning and Design (1 Month)**

Week 1–2: Requirement Gathering & Planning

- Define scope and objectives of the e-commerce platform.
- Study existing online retail platforms and shortlist key features.
- Create a task timeline for UI/UX, backend logic, and database schemas.

Week 3–4: UI/UX Design

- Design responsive user interfaces using Figma.
- Create mockups for the homepage, product list, product detail, signup/login, cart, and checkout pages.
- Finalize backend API structure for product, cart, and order modules.

- **Phase 2: Development (2 Months)**

Week 5–6: Flutter Frontend Development

- Develop homepage with dynamic product tiles.
- Implement login/signup pages with Firebase Authentication.
- Create reusable product and cart widgets for modular development.

Week 7–10: Backend & Database Integration

- Develop REST APIs using Node.js or PHP connected with MySQL.
- Handle user management, cart logic, and checkout processing.
- Integrate Firebase real-time updates for cart and order confirmation.

Week 11–12: Testing and Debugging

- Conduct unit testing for cart logic and SQL queries.
- Perform UI testing on multiple browsers and screen sizes.
- Validate authentication flows, order placement, and data consistency.

- **Phase 3: Deployment & Launch (1 Month)**

Week 13–14: Final Integration and Deployment

- Deploy the web app using flutter build web.
- Upload to Netlify and configure routing and analytics.
- Perform final QA, user testing, and prepare for launch.

1.4 Organization of the Report

Modernizing Online Shopping: An Overview of the Flutter-Based E-Commerce Solution

1. **Introduction**
   - ✓ 1.1 Project Background
   - ✓ 1.2 Objective of the Project
   - ✓ 1.3 Report Structure
2. **Understanding the Platform**
   - ✓ 2.1 Key Stakeholders: Users, Admins, Vendors
3. **Core Features**
   - ✓ 3.1 Firebase Authentication
     - ▪ Role in User Security and Login Flow
   - ✓ 3.2 Checkout & Order Tracking
     - ▪ Transparent Summary and History
     - ▪ Firebase Notifications and Alerts
   - ✓ 3.3 Admin Tools & Analytics
     - ▪ Google Analytics Reports
     - ▪ Firebase Real-Time Dashboard
4. **Platform Benefits**
   - ✓ 4.1 Seamless User Experience
   - ✓ 4.2 Secure and Scalable Architecture

CHAPTER 2.
LITERATURE REVIEW/BACKGROUND STUDY

The rapid evolution of digital technologies has significantly influenced the development of modern e-commerce platforms. With the increasing demand for seamless, secure, and responsive online shopping experiences, developers are turning to cross-platform frameworks like Flutter, combined with cloud services such as Firebase and relational databases like SQL. These technologies collectively address challenges related to authentication, data management, real-time interactions, and scalability. This literature review explores existing research and solutions in the domain of web-based e-commerce applications, focusing on the integration of Flutter for UI development, Firebase for authentication and analytics, and SQL for backend data operations. By analyzing current approaches, functional capabilities, and limitations, this review highlights how modern frameworks are being used to streamline online retail workflows, improve user engagement, and enable scalable e-commerce solutions for businesses of all sizes.

2.1 Existing Solutions

Numerous digital platforms have emerged to facilitate online shopping and streamline e-commerce operations. Among these, industry leaders such as Amazon, Flipkart, and Shopify offer powerful systems that enable smooth customer experiences, including secure user authentication, extensive product listings, cart management, checkout systems, and order tracking. These platforms have set high standards for functionality and performance in the digital commerce domain.However, many of these platforms are designed for large enterprises and often require substantial technical expertise, infrastructure, and investment. This creates a gap for small and medium-sized businesses looking for customizable, affordable, and scalable alternatives. These businesses need solutions that are easy to manage and can be tailored to specific business models without compromising functionality. In this landscape, the **Flutter-Based E-Commerce Web Application** stands out as a versatile and modern solution.

Built using the Flutter framework, this platform leverages Firebase for real-time authentication and SQL for structured data management, enabling efficient order processing, cart tracking, product browsing, and administrative functions. It empowers developers and vendors to launch scalable and secure e-commerce sites with reduced complexity.

2.1 Bibliometric Analysis

This bibliometric analysis draws from databases such as IEEE Xplore, ACM Digital Library, Scopus, and Google Scholar. Search terms like "Flutter e-commerce web app," "Firebase authentication integration," and "SQL database in Flutter" were used to filter relevant literature. The review includes journal articles, conference papers, case studies, and white papers related to modern web-based e-commerce solutions and Flutter application development.

Results:

Adoption and Impact

• Studies highlight the increasing adoption of Flutter for web app development due to its cross-platform support and rapid UI rendering capabilities.

• Firebase's user authentication module is praised for secure login processes, scalability, and ease of integration with frontend frameworks like Flutter.

Functional Features

• Research indicates that integrating Firebase and SQL in Flutter enhances real-time interactions such as cart updates, order tracking, and checkout processing.

• User interface improvements and fast state management tools (like Provider or Riverpod) have shown positive effects on usability and performance.

Future Directions

• Scholars suggest incorporating artificial intelligence for personalized product recommendations, AI chatbots for customer support, and blockchain for order verification.

• Ensuring accessibility for all users, including those with disabilities, is emphasized as a future development focus.

2.1 Goles / Objective

Goal:

To build a secure, scalable, and user-friendly e-commerce platform that allows seamless shopping, real-time cart management, efficient order processing, and user authentication using Flutter, Firebase, and SQL.

Objectives:

- ✓ **Streamlined Shopping Experience**

  Design an intuitive, fast-loading UI in Flutter for seamless product discovery. Incorporate filters, search, and detail views to simplify product selection for end-users.

- ✓ **Efficient Authentication**

  Implement Firebase Authentication to support email/password login. Enable secure registration and sign-in workflows while protecting user data.

- ✓ **Real-Time Cart & Wishlist Management**

  Allow users to add, edit, or remove products from their cart and wishlist with real-time sync using Firebase. Maintain state with Provider for a smooth user experience.

- ✓ **Dynamic Product Handling via SQL**

  Enable vendors to manage product listings, categories, and inventory. Use SQL to perform CRUD operations and run queries for admin panels and dashboards.

- ✓ **Secure Checkout & Order History**

  Provide an easy-to-follow checkout process. Save order summaries to SQL and allow users to view past orders. Implement auto-generated order IDs for easy tracking.

- ✓ **Event Logging & Analytics**

  Use Firebase Analytics to log events like “Product Viewed” or “Checkout Started.” Help admins track user flow and identify trends to enhance the platform.

- ✓ **Deployment and Accessibility**

  Deploy the application on Netlify or Firebase Hosting. Ensure responsive design, quick build times, and wide accessibility on both desktop and mobile browsers.

CHAPTER 3.

DESIGN FLOW/PROCESS

3.1 Design Constraints

Design Constraints for Helping Hand:

1. Security and Privacy: Ensuring the security and privacy of users' personal information, financial data, and communication channels is paramount. The platform must adhere to industry standards and regulations regarding data protection and encryption.

2. Scalability: The platform should be designed to handle a large volume of users, donations, and activities without compromising performance. It should be scalable to accommodate growth and increasing demand over time.

3.2 ER- Diagram

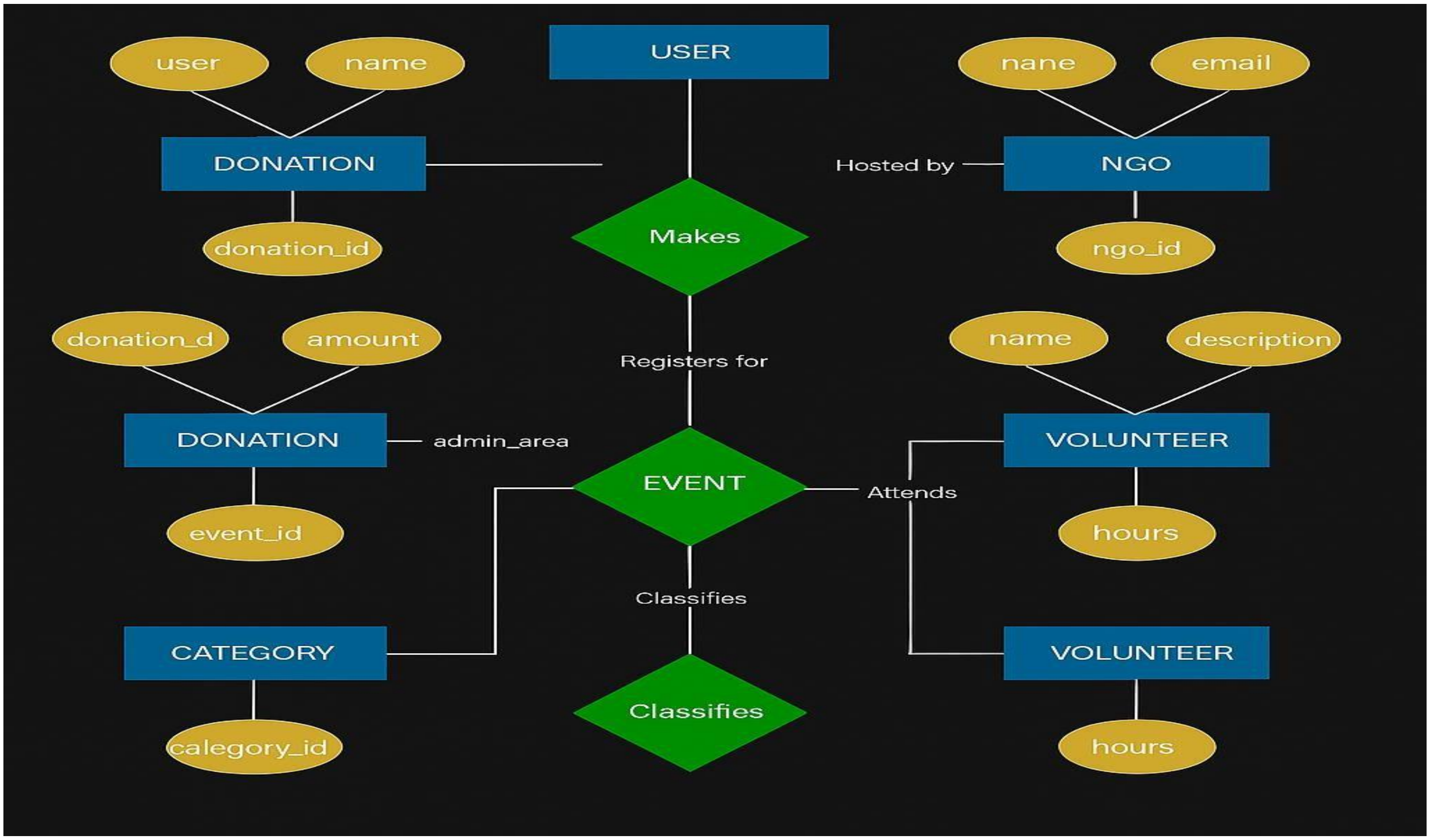
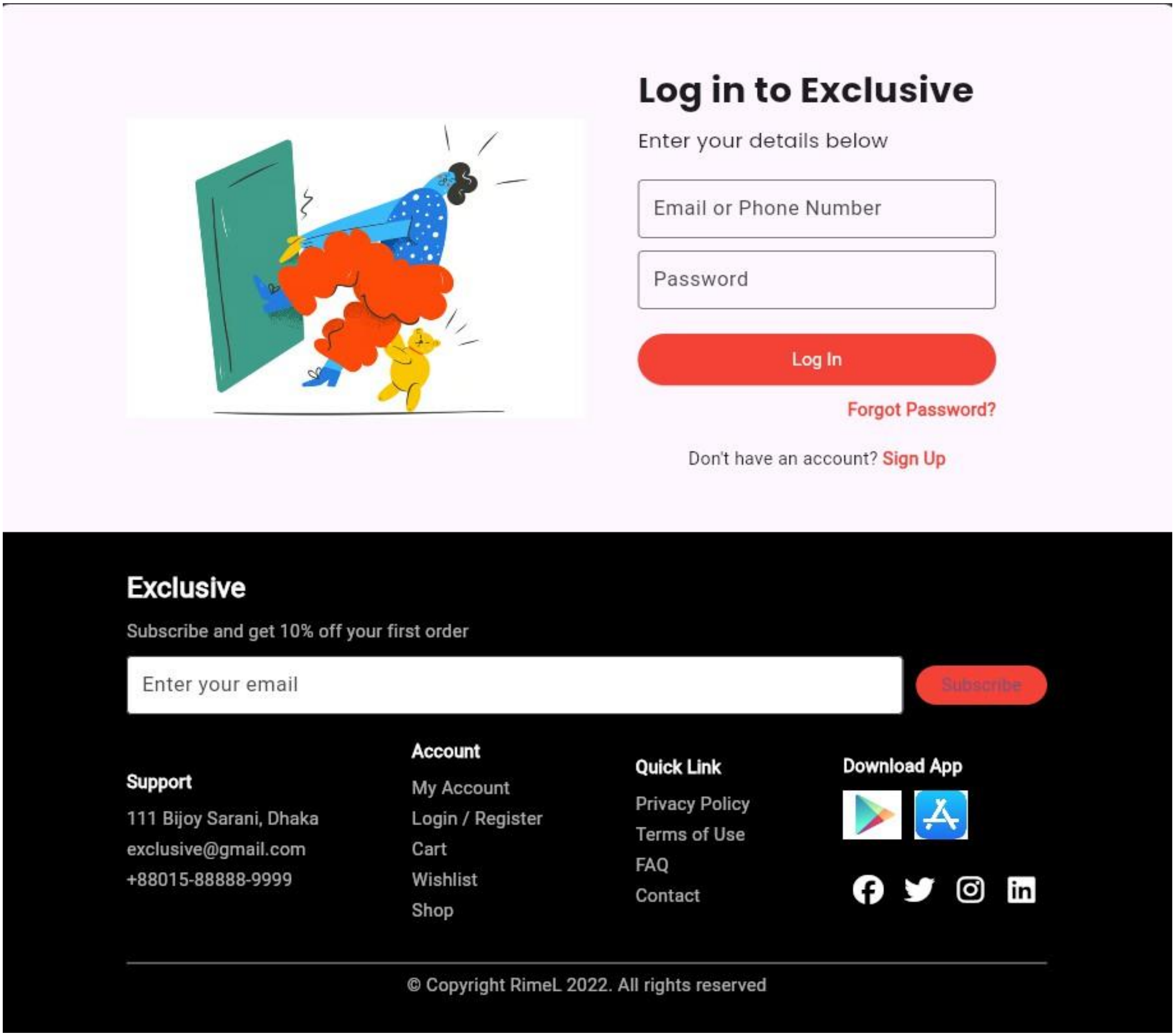


Figure 1.1

CHAPTER 4.

CODE OF PROJECT

4.1 LOGIN PAGE CODE



Log in to Exclusive

Enter your details below

Email or Phone Number

Password

Don't have an account? Sign Up

Exclusive

Subscribe and get 10% off your first order

Enter your email

Support

111 Bijoy Sarani, Dhaka
exclusive@gmail.com
+88015-88888-9999

© Copyright RimeL 2022. All rights reserved

Figure 1.2

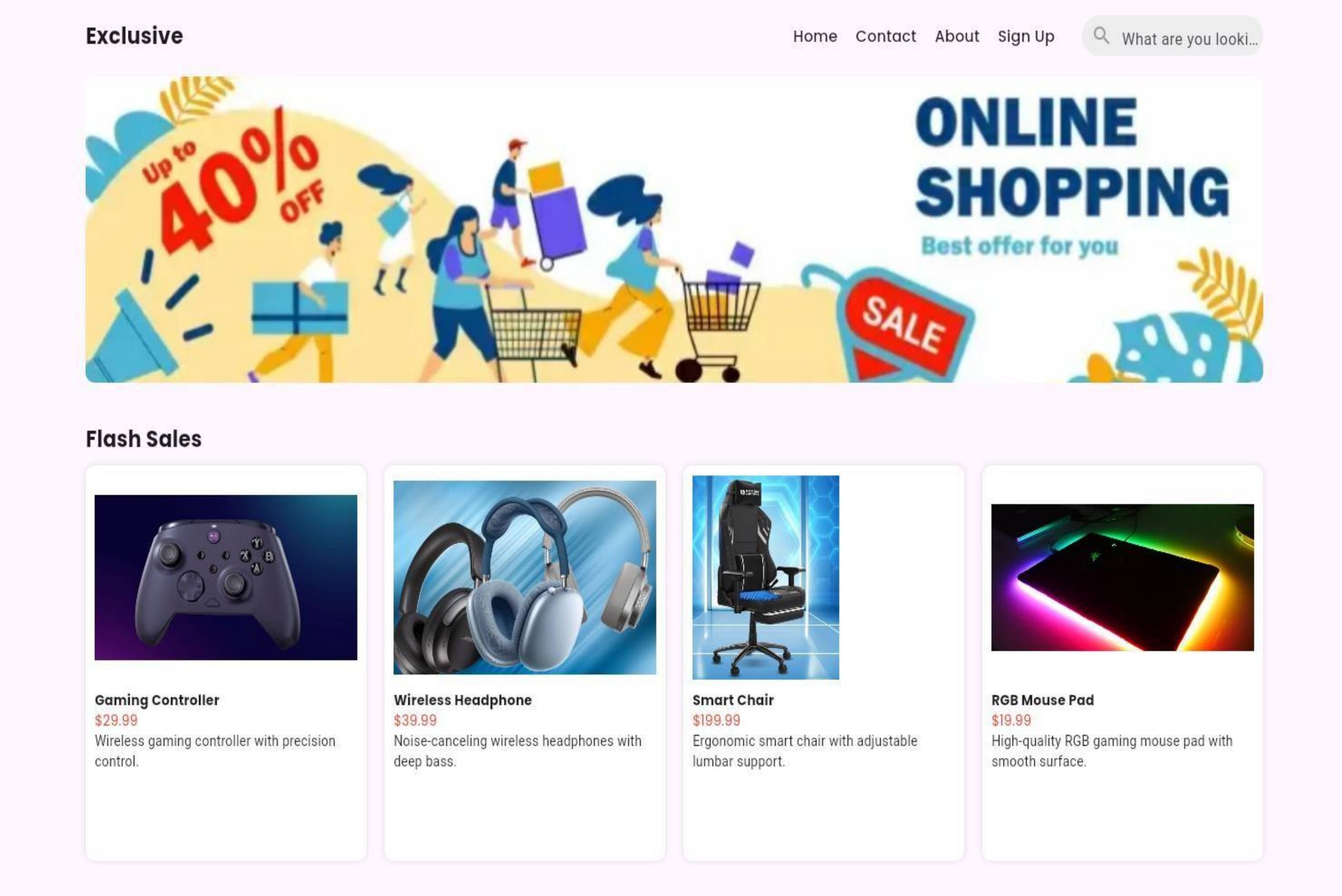
```
import essential files*

  class LoginScreen extends StatelessWidget {
  final TextEditingController emailController = TextEditingController();
  final TextEditingController passwordController = TextEditingController();
  // Firebase Login Function
  Future<void> _login(BuildContext context) async {
    try {
      await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: emailController.text.trim(),
        password: passwordController.text.trim(),
      );
      Navigator.pushNamed(context, AppRoutes.home); // Redirect on success
    } on FirebaseAuthException catch (e) {
      ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text(e.message ?? "Login failed")),
      );
    }
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Column(
          children: [
            // Top Section - Image & Login Form
            Container(
              padding: EdgeInsets.symmetric(horizontal: 100, vertical: 50),
              child: Row(
                children: [
                  // Left Image
                  Expanded(
                    child: Image.asset(
                      "assets/images/login_illustration.png",
                      fit: BoxFit.contain,
                    ),
                  ),
                  // Right Form Section
                  Expanded(
                    child: Padding(
                      padding: const EdgeInsets.symmetric(horizontal: 40),
                      child: Column(
                        crossAxisAlignment: CrossAxisAlignment.start,
                        children: [
                          Text(
                            "Log in to Exclusive",
                            style: GoogleFonts.poppins(
                              fontSize: 28,
                              fontWeight: FontWeight.bold,
  ),
),
SizedBox(height: 10),
Text(
  "Enter your details below",
  style: GoogleFonts.poppins(fontSize: 16),
),
SizedBox(height: 20),
// Email Input
TextField(
  controller: emailController,
  decoration: InputDecoration(
  labelText: "Email or Phone Number",
  border: OutlineInputBorder(),
  ),
),
SizedBox(height: 10),
// Password Input
TextField(
  controller: passwordController,
  obscureText: true,
  decoration: InputDecoration(
    labelText: "Password",
    border: OutlineInputBorder(),
  ),
),
SizedBox(height: 20),
// Login Button
ElevatedButton(
  onPressed: () => _login(context),
  style: ElevatedButton.styleFrom(
  backgroundColor: Colors.red,
    minimumSize: Size(double.infinity, 50),
  ),
  child: Text("Log In", style: TextStyle(color: Colors.white)),
),
SizedBox(height: 10),
Align(
  alignment: Alignment.centerRight,
  child: GestureDetector(
    onTap: () {
      // Navigate to Forgot Password Page (optional)
    },
    child: Text(
      "Forgot Password?",
      style: TextStyle(
      color: Colors.red,
        fontWeight: FontWeight.bold,
      ),
    ),
                ),
              ),
              SizedBox(height: 20),
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text("Don't have an account? "),
                  GestureDetector(
                    onTap: () {
                      Navigator.pushNamed(context, AppRoutes.signup);
                    },
                    child: Text(
                      "Sign Up",
                      style: TextStyle(
                        color: Colors.red,
                        fontWeight: FontWeight.bold,
                      ),
                    ),
                  ),
                ],
              ),
            ],
          ),
        ),
      ),
    ],
  ),
),

// Footer Section
Container(
  width: double.infinity,
  color: Colors.black,
  padding: EdgeInsets.symmetric(vertical: 30, horizontal: 100),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text("Exclusive",
          style: TextStyle(fontSize: 22, color: Colors.white, fontWeight: FontWeight.bold)),
      SizedBox(height: 10),
      Text("Subscribe and get 10% off your first order",
          style: TextStyle(color: Colors.white70)),
      SizedBox(height: 10),
      Row(
        children: [
          Expanded(
            child: TextField(
              decoration: InputDecoration(
                filled: true,
                fillColor: Colors.white,
          hintText: "Enter your email",
          border: OutlineInputBorder(),
        ),
      ),
    ),
    SizedBox(width: 10),
    ElevatedButton(
    onPressed: () {},
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.red,
        padding: EdgeInsets.symmetric(horizontal: 20),
      ),
      child: Text("Subscribe"),
    ),
  ],
),
SizedBox(height: 20),

Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    _footerColumn("Support", [
      "111 Bijoy Sarani, Dhaka",
      "exclusive@gmail.com",
      "+88015-88888-9999",
    ]),
    _footerColumn("Account", [
      "My Account",
      "Login / Register",
      "Cart",
      "Wishlist",
      "Shop",
    ]),
    _footerColumn("Quick Link", [
      "Privacy Policy",
      "Terms of Use",
      "FAQ",
      "Contact",
    ]),
    Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text("Download App", style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold)),
        SizedBox(height: 10),
        Row(
          children: [
            Image.asset("assets/images/playstore.png", height: 40),
            SizedBox(width: 10),
            Image.asset("assets/images/appstore.png", height: 40),
          ],
                          ),
                          SizedBox(height: 20),
                          Row(
                            children: [
                              IconButton(icon: FaIcon(FontAwesomeIcons.facebook, color: Colors.white), onPressed: ()
                              {}),
                              IconButton(icon: FaIcon(FontAwesomeIcons.twitter, color: Colors.white), onPressed: () {}),
                              IconButton(icon: FaIcon(FontAwesomeIcons.instagram, color: Colors.white), onPressed: ()
                          {}),
                              IconButton(icon: FaIcon(FontAwesomeIcons.linkedin, color: Colors.white), onPressed: ()
                          {}),
                            ],
                          ),
                        ],
                      ),
                    ],
                  ),
                  SizedBox(height: 20),
                  Divider(color: Colors.white70),
                  Center(
                    child: Text(
                      "© Copyright RimeL 2022. All rights reserved",
                      style: TextStyle(color: Colors.white70),
                    ),
                  ),
                ],
              ),
            ),
          ],
        ),
      ),
    );
  }
  Widget _footerColumn(String title, List<String> items) {
    return Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(title, style: TextStyle(color: Colors.white, fontWeight: FontWeight.bold)),
        SizedBox(height: 10),
        for (var item in items)
          Padding(
            padding: const EdgeInsets.only(bottom: 5),
            child: Text(item, style: TextStyle(color: Colors.white70)),
          ),
      ],
    );
  }
}
```

4.2 HOME PAGE CODE



Gaming Controller
$29.99
Wireless gaming controller with precision control.

Wireless Headphone
$39.99
Noise-canceling wireless headphones with deep bass.

Smart Chair
$199.99
Ergonomic smart chair with adjustable lumbar support.

RGB Mouse Pad
$19.99
High-quality RGB gaming mouse pad with smooth surface.

Figure 1.3

```
import necessary files*

class HomeScreen extends StatelessWidget {
  final List<Map<String, dynamic>> products = [
    {
      "image": "assets/images/product1.png",
      "title": "Gaming Controller",
      "price": "29.99",
      "description": "Wireless gaming controller with precision control."
    },
    {
      "image": "assets/images/product2.png",
    "title": "Wireless Headphone",
    "price": "39.99",
    "description": "Noise-canceling wireless headphones with deep bass."
  },
  {
    "image": "assets/images/product3.png",
    "title": "Smart Chair",
    "price": "199.99",
    "description": "Ergonomic smart chair with adjustable lumbar support."
  },
  {
    "image": "assets/images/product4.png",
    "title": "RGB Mouse Pad",
    "price": "19.99",
    "description": "High-quality RGB gaming mouse pad with smooth surface."
  },
  {
    "image": "assets/images/product5.png",
    "title": "Stylish Handbag",
    "price": "49.99",
    "description": "Premium leather handbag with multiple compartments."
  },
  {
    "image": "assets/images/product6.png",
    "title": "Smartwatch Pro",
    "price": "199.99",
    "description": "Advanced smartwatch with heart rate & GPS tracking."
  },
  {
    "image": "assets/images/product7.png",
    "title": "Wireless Speaker",
    "price": "89.99",
    "description": "Portable Bluetooth speaker with high-quality sound."
  },
  {
    "image": "assets/images/product8.png",
    "title": "Professional Camera",
    "price": "499.99",
    "description": "DSLR camera with high-resolution lens and 4K video."
  },
];

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
    children: [
     _buildHeader(context),

     // Banner Section
     _banner(),

     SizedBox(height: 30),

     // Flash Sales Section
     _sectionTitle("Flash Sales"),
     _productGrid(context, products.sublist(0, 4)),

     SizedBox(height: 30),

     // Browse by Category
     _sectionTitle("Browse By Category"),
     _categoryBrowse(),

     SizedBox(height: 30),

     // Best Selling Products
     _sectionTitle("Best Selling Products"),
     _productGrid(context, products),

     SizedBox(height: 30),

     // Promotional Banner
     _promoBanner(),

     SizedBox(height: 30),

     // Explore Our Products
     _sectionTitle("Explore Our Products"),
     _productGrid(context, products),

     SizedBox(height: 30),

     // New Arrivals
     _sectionTitle("New Arrivals"),
     _newArrivals(),

     SizedBox(height: 50),
    ],
   ),
  ),
 );
}
```

```
Widget _buildHeader(BuildContext context) {
  return Container(
    padding: EdgeInsets.symmetric(horizontal: 100, vertical: 20),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      children: [
        Text(
          "Exclusive",
          style: GoogleFonts.poppins(
            fontSize: 22,
            fontWeight: FontWeight.bold,
          ),
        ),
        Row(
          children: [
            _navItem(context, "Home", AppRoutes.home),
            _navItem(context, "Contact", AppRoutes.contact),
            _navItem(context, "About", AppRoutes.about),
            _navItem(context, "Sign Up", AppRoutes.signup),
            SizedBox(width: 20),
            _searchBar(),
          ],
        ),
      ],
    ),
  );
}

Widget _banner() {
  return Container(
    margin: EdgeInsets.symmetric(horizontal: 100),
    height: 300,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      image: DecorationImage(
        image: AssetImage("assets/images/banner.png"),
        fit: BoxFit.cover,
      ),
    ),
  );
}

Widget _navItem(BuildContext context, String title, String route) {
  return Padding(
    padding: const EdgeInsets.symmetric(horizontal: 10),
    child: GestureDetector(
      onTap: () {
        Navigator.pushNamed(context, route);
      },
      child: Text(
        title,
        style: GoogleFonts.poppins(fontSize: 16, fontWeight: FontWeight.w500),
      ),
    ),
  );
}

Widget _searchBar() {
  return Container(
  width: 200,
    height: 40,
    decoration: BoxDecoration(
      color: Colors.grey[200],
      borderRadius: BorderRadius.circular(20),
    ),
    child: Row(
      children: [
      Padding(
          padding: EdgeInsets.symmetric(horizontal: 10),
          child: Icon(Icons.search, color: Colors.grey),
        ),
        Expanded(
          child: TextField(
            decoration: InputDecoration(
              hintText: "What are you looking for?",
              border: InputBorder.none,
            ),
          ),
        ),
      ],
    ),
  );
}

Widget _sectionTitle(String title) {
  return Padding(
    padding: EdgeInsets.symmetric(horizontal: 100, vertical: 10),
    child: Text(
      title,
      style: GoogleFonts.poppins(fontSize: 22, fontWeight: FontWeight.bold),
    ),
  );
}

Widget _productGrid(BuildContext context, List<Map<String, dynamic>> products) {
  return Container(
    padding: EdgeInsets.symmetric(horizontal: 100),
    child: GridView.builder(
      shrinkWrap: true,
      physics: NeverScrollableScrollPhysics(),
      itemCount: products.length,
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 4,
        crossAxisSpacing: 20,
        mainAxisSpacing: 20,
        childAspectRatio: 0.8,
      ),
      itemBuilder: (context, index) {
        return _productCard(context, products[index]);
      },
    ),
  );
}

Widget _productCard(BuildContext context, Map<String, dynamic> product) {
  return GestureDetector(
    onTap: () {
      Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => ProductDetailScreen(product: product),
        ),
      );
    },
    child: Container(
      padding: EdgeInsets.all(10),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        color: Colors.white,
        boxShadow: [BoxShadow(color: Colors.black12, blurRadius: 5)],
      ),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Image.asset(product["image"], height: 200),
          SizedBox(height: 10),
          Text(product["title"], style: GoogleFonts.poppins(fontSize: 14, fontWeight: FontWeight.bold)),
          Text("\$${product["price"]}", style: GoogleFonts.poppins(fontSize: 14, color: Colors.red)),
          Text(product["description"], maxLines: 2, overflow: TextOverflow.ellipsis),
        ],
      ),
    ),
  );
}
```

```
Widget _categoryBrowse() {
  return Container(
    padding: EdgeInsets.symmetric(horizontal: 100),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        _categoryCard("assets/images/laptop.png",  "Laptops"),
        _categoryCard("assets/images/mobile.png",  "Mobiles"),
        _categoryCard("assets/images/camera.png",  "Cameras"),
        _categoryCard("assets/images/headphones.png",  "Headphones"),
      ],
    ),
  );
}

Widget _categoryCard(String icon, String title) {
  return Column(
    children: [
      Image.asset(icon, height: 150),
      SizedBox(height: 10),
      Text(title, style: GoogleFonts.poppins(fontSize: 16, fontWeight: FontWeight.w500)),
    ],
  );
}

Widget _promoBanner() {
  return Container(
    margin: EdgeInsets.symmetric(horizontal:  100),
    height: 250,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      image: DecorationImage(
        image: AssetImage("assets/images/promo.png"),
        fit: BoxFit.fitWidth,
      ),
    ),
  );
}

Widget _newArrivals() {
  return Container(
    padding: EdgeInsets.symmetric(horizontal: 100),
    child: Row(
      children: [
        Expanded(
      child: Image.asset("assets/images/newrarrival1.png", fit: BoxFit.cover),
     ),
     SizedBox(width: 20),
     Expanded(
      child: Image.asset("assets/images/newarrival2.png", fit: BoxFit.cover),
     ),
    ],
   ),
  );
 }
}
```

CHAPTER 5.
TESTING

5. 1 Testing Objective

Additionally, identifying and resolving any bugs or issues is paramount to maintaining the platform's integrity and usability. Regular debugging processes help in enhancing the user experience and preventing disruptions in service.

Furthermore, verifying the security and stability of the application is crucial in safeguarding sensitive data and ensuring continuous operation without vulnerabilities. Regular security assessments and updates are essential to mitigate potential risks.

Moreover, assessing the user interface for usability and accessibility is vital for ensuring an inclusive experience for all users. User feedback and usability testing can provide insights into areas for improvement to enhance user satisfaction and engagement.

5.1.1 Testing Type

Functional Testing: To verify that all functions of the application work according to specifications.

User Acceptance Testing (UAT): Involving real users to assess whether the application meets their needs and expectations.

Performance Testing: To evaluate the application's response time, scalability, and resource usage under various conditions.
Usability Testing: To assess the user interface for intuitiveness and ease of use.

Accessibility Testing: Ensuring the application is accessible to users with disabilities.

5.1 Test Environment

Development Environment: Local development machines with version control (Git), and integrated development environment (IDE).

Testing Environment: Staging server to mimic the production environment for testing purposes.

CHAPTER 6.
FUTURE SCOPE & CONCLUSION

6.1 Future Enhancement

Implementing advanced analytics and reporting features will furnish invaluable insights into donation trends, volunteer engagement, and event effectiveness. These insights can inform decision-making, enabling more targeted and impactful initiatives.

Furthermore, integrating machine learning algorithms stands to revolutionize user experiences by personalizing interactions, recommending pertinent donation opportunities, and optimizing fundraising strategies. Such customization can enhance user engagement and foster a deeper connection to the platform's mission.

Equally crucial is the enhancement of security measures to fortify user data and financial transactions against potential threats and vulnerabilities. Strengthening security protocols instills trust among users, assuring them of the platform's commitment to their privacy and safety.

Finally, expanding the application's scalability and performance is essential to accommodate a burgeoning user base and escalating data volume.

6.2 CONCLUSION

The Flutter-Based E-Commerce Web Application marks a pivotal advancement in streamlining the online shopping experience through intuitive design, modern technologies, and seamless integration. Developed using Flutter for web, this platform offers users a responsive, engaging, and efficient interface for browsing, purchasing, and managing products with ease. By incorporating dynamic product management, secure authentication, cart functionality, checkout processing, and user-specific operations such as wishlist and profile management, the platform bridges the gap between sellers and customers in a transparent and user-friendly manner. The use of Firebase for authentication and cloud storage, as well as plans for integrating SQL for robust data handling, reflects the project’s commitment to security, scalability, and data integrity. Looking ahead, the continuous refinement and expansion of the system will open new possibilities for business scalability, customer satisfaction, and global accessibility.

REFERENCES

[1]. M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, “Blockchain technologies for the Internet of Things: Research issues and challenges,” IEEE Internet Things J., vol. 6, no. 2, pp. 2188–2204, 2019, doi: 10.1109/JIOT.2018.2882794.

[2]. H. Saleh, S. Avdoshin and A. Dzhonov, "Platform for Tracking Donations of Charitable Foundations Based on Blockchain Technology," 2019 Actual Problems of Systems and Software Engineering (APSSE), Moscow, Russia, 2019, pp. 182-187, doi: 10.1109/APSSE47353.2019.00031.

[3]. N. Alblihed, M. Almutairi, R. Almahmoud, S. Aladhadh and A. Alabdulaf, "Developing Food Charity Operations Management System," 2022 2nd International Conference on Computing and Information Technology (ICCIT), Tabuk, Saudi Arabia, 2022, pp. 93-96, doi: 10.1109/ICCIT52419.2022.9711609.

[4]. S. P. Tiwari, G. Phonsa and P. Singh, "Study and Comparative analysis of Donation based websites," 2021 International Conference on Computing Sciences (ICCS), Phagwara, India, 2021, pp. 202-205, doi: 10.1109/ICCS54944.2021.00047.

[5]. W. Lee, D. Kim and B. R. Jeon, "A Study on Blockchain Application in Donation Platform," 2021 21st ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter), Ho Chi Minh City, Vietnam, 2021, pp. 284-286, doi: 10.1109/SNPDWinter52325.2021.00075.

APPENDIX

Flutter-Based E-Commerce Web Application

1. User Interfaces

- **Customer Interface**:
  Customers can sign up, log in, browse products, add items to their cart or wishlist, manage their profile, track order history, and proceed to checkout through a user-friendly web interface.

2. Technologies Used

- **Frontend**:
  Flutter Web using Dart for building responsive and interactive UI components.

- **Authentication**:
  Firebase Authentication for secure login using email/password.

3. Database Schema

- **Users**:
  Stores customer details like name, email, password (securely hashed), and profile information.

- **Products**:
  Contains product titles, descriptions, prices, stock count, images, and categories.

4. Security Measures

- **Secure Authentication**:
  Firebase Authentication ensures secure login and session management.