Index

|  |  |  |
| --- | --- | --- |
| **Sr .No** | **Contents** | **Page No** |
| **1** | **Introduction** |  |
|  | 1.1 Problem definition |  |
|  | 1.2 Existing system |  |
|  | 1.3 Need for Computerization |  |
|  | 1.4 Scope of the proposed System |  |
|  |  |  |
| **2** | **TECHNOLOGY** |  |
|  | 2.1 Technology used |  |
|  | 2.2 Platform with version details |  |
|  | 2.3 Tools used |  |
|  |  |  |
| **3** | **ANALYSIS & DESIGN**  **Object Oriented Approach** |  |
|  | 3.1 Activity Diagram |  |
|  | 3.2 Workflow |  |
|  | 3.3 Flow Diagram |  |
|  |  |  |
| **5** | **Testing Strategies** |  |
|  |  |  |
| **6** | **DRAWBACKS** |  |
|  |  |  |
| **7** | **CONCLUSION** |  |
|  |  |  |
| **8** | **REFERENCE** |  |

**Introduction**

The DessertDeck (Sweet Shop Management system) is a comprehensive web application designed to manage an online Sweet Shop. It allows customers to browse and purchase a variety of sweets, while providing an interface for administrators to manage products, orders, and customer information. The application is built using Django, Python web framework.

Features

* User Registration:
  + Users can create accounts, log in, and log out.
  + Password management and recovery.
* Product Catalogue:
  + Display of products with details such as name, description, price, and image.
  + Search and filter products.
* Shopping Cart:
  + Add, update, and remove items.
  + View total price.
* Order Placement and Tracking:
  + Place orders and receive confirmation.
  + Track order status.
* User Profile Management:
  + Update personal information and view order history.
* Admin Panel:
  + Manage products (add, update, delete).
  + Manage orders (view, update status).
  + Manage users (view, update, delete accounts)

**Problem definition**

The Sweet Shop is a traditional retail business specializing in the sale of various Sweets. With the growing trend of online shopping and the need to expand the business reach beyond the local community, there is a need to develop a web-based platform to facilitate online sales and streamline business operations. This platform will allow customers to browse the product catalog, place orders online, and manage their profiles, while also enabling administrators to manage inventory, orders, and customer information efficiently.

The primary objective of this project is to develop an online sweet shop web application that addresses the following needs:

Technical Requirements:

* Scalability: Design the application to handle a growing number of users and transactions as the business expands.
* Performance: Ensure the application performs efficiently, with fast loading times and minimal downtime.
* Maintainability: Develop the application using best practices and clean code to facilitate easy maintenance and future enhancements.

**Existing system**

**Manual Systems:** Many sweet shops still rely on traditional manual methods for inventory management, sales tracking, and customer management. These systems involve handwritten records, manual calculations, and paper-based processes.

**Basic Digital Systems**: Some sweet shops utilize basic digital tools such as spreadsheets (e.g., Excel) or simple software applications for inventory tracking and sales management. These systems offer limited automation and may lack advanced features like integration with POS systems or CRM functionalities.

**Need for Computerization**

Computerizing a sweet shop management system offers numerous benefits that streamline operations, enhance efficiency, and improve customer experience. Here are some key points highlighting the need for computerization:

* **Real-Time Tracking**: Automated systems allow real-time tracking of inventory, preventing stockouts or overstock situations.
* **Accurate Records**: Keeps accurate records of ingredients, products, and materials, reducing wastage and ensuring freshness.
* **Automated Accounting**: Keeps track of expenses, revenues, and profits with minimal manual intervention.
* **Online Ordering**: Integration with online ordering platforms to cater to the growing demand for online shopping.
* Convenient Shopping: Allow customers to browse a wide range of sweet products, view details, and make purchases from the comfort of their homes.

Computerizing a sweet shop management system transforms the way the business operates, leading to enhanced efficiency, better financial management, improved customer satisfaction, and scalability for future growth. Implementing such a system is a strategic investment that can drive long-term success

**Scope of the proposed System**

The scope of the proposed computerized sweet shop management system includes:

* Inventory Management: Real-time tracking, automatic reordering, and expiry date management.
* Sales and Billing: Efficient POS system, detailed sales reporting, and management of discounts and promotions.
* Customer Relationship Management (CRM): Customer database, loyalty programs, and feedback collection.
* Order Management: Integration with online ordering, real-time order tracking, and handling custom orders.
* Financial Management: Expense tracking, revenue management, and generating financial reports.
* User Interface and Experience: User-friendly interface, multi-device compatibility, and multi-language support.

This system aims to enhance efficiency, improve customer satisfaction, and support business growth.

**TECHNOLOGY**

**Technology used**

* Python 3.12.3 or later
* Django 3.0 or later
* SQLite (default database for Django projects)
* pip (Python package installer)

**Tools used**

* **Programming Languages:**

**- Backend: Python, Java, PHP for server-side logic.**

**-Frontend : HTML, CSS, JavaScript for building user interfaces.**

* **Frameworks and Libraries:**

**- Backend: Django, Flask (Python),**

**- Frontend: React, Angular, Vue.js for dynamic and responsive front-end development.**

* **Database Management Systems:**

**- Relational Databases: SQLite**

* **Development Environment:**

**- IDEs: Visual Studio Code, PyCharm, IntelliJ IDEA for coding and debugging.**

* **Security Tools:**

**- Encryption: SSL/TLS for secure data transmission.**

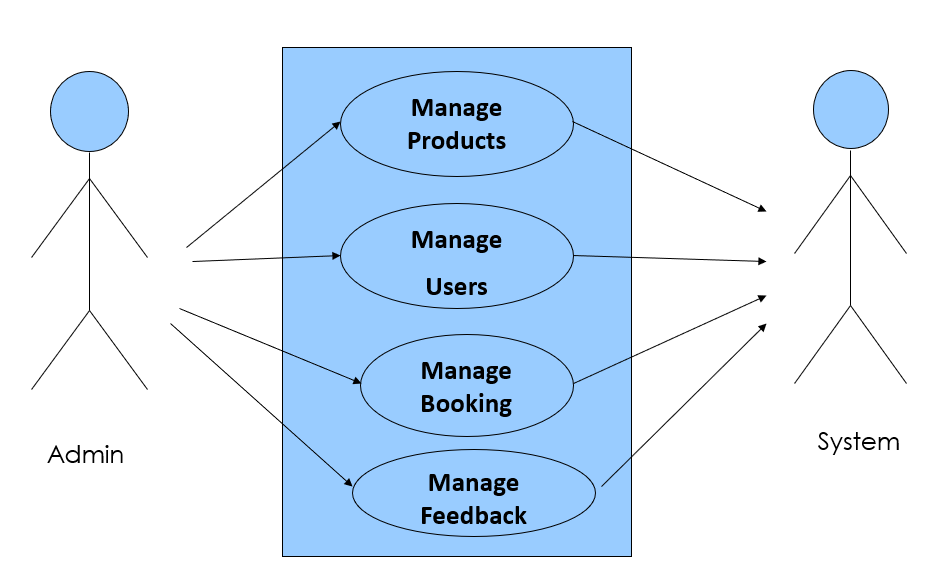
**These technical tools collectively support the development, deployment, and maintenance of the computerized Sweet Shop management system, ensuring it is robust, scalable, and efficient.**

**ANALYSIS & DESIGN**

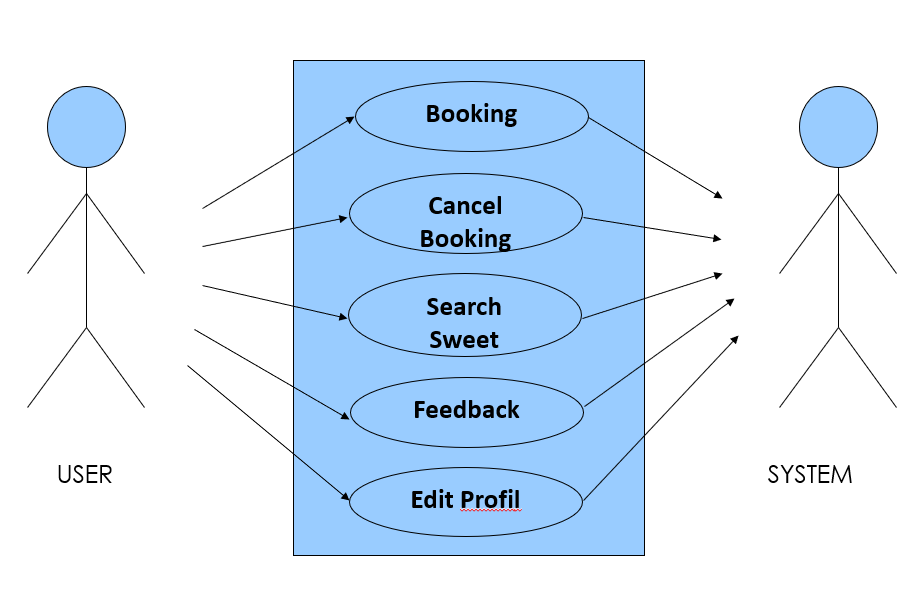
*Contents:*

* Use case diagram
* Sequence Diagram
* Data flow diagram
* ER diagram

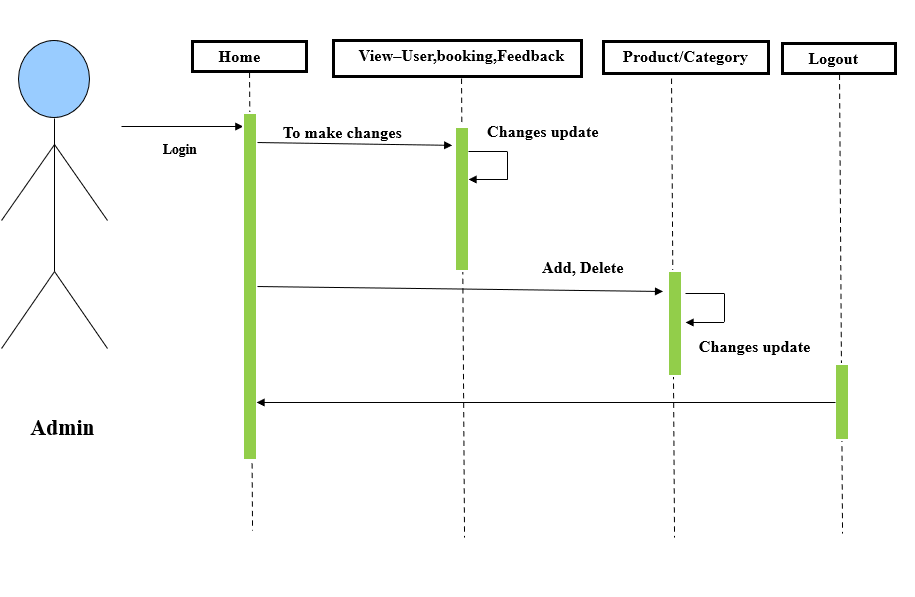
**Use Case Diagram:**

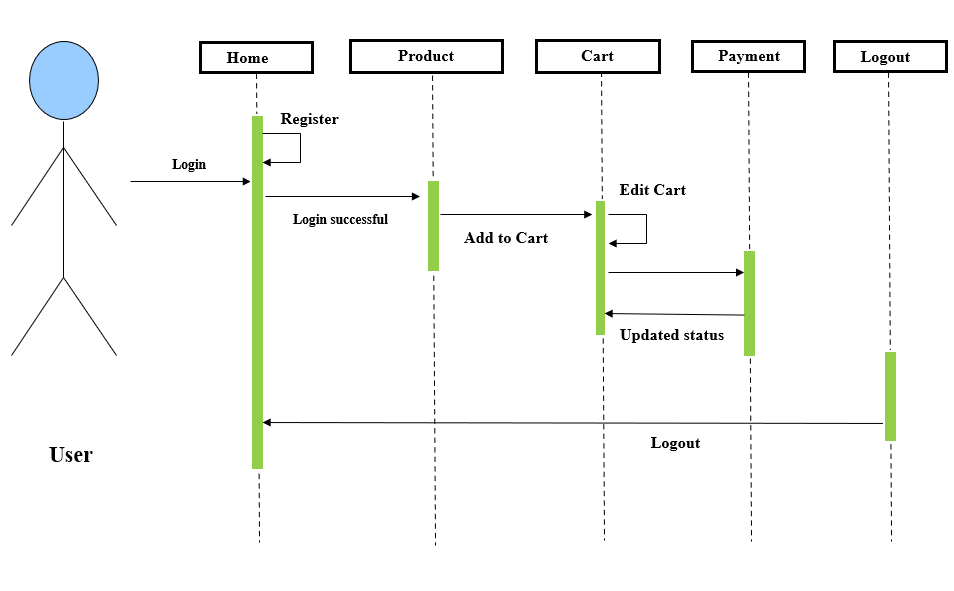
**Use Case Diagram between ADMIN and SYSTEM**

**Use Case Diagram between USER and SYSTEM:**

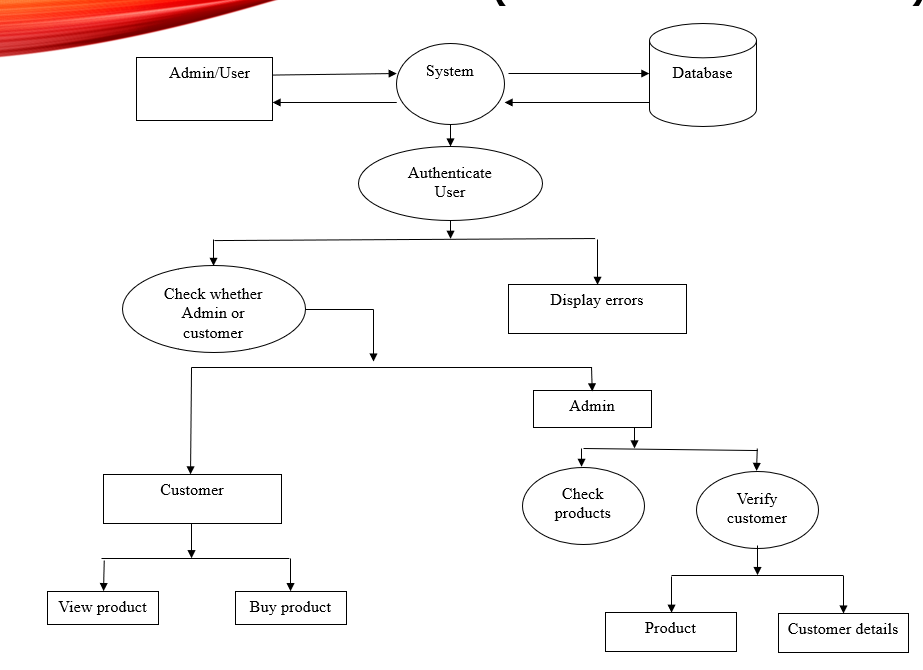
****

**Sequence Diagram (Admin)**

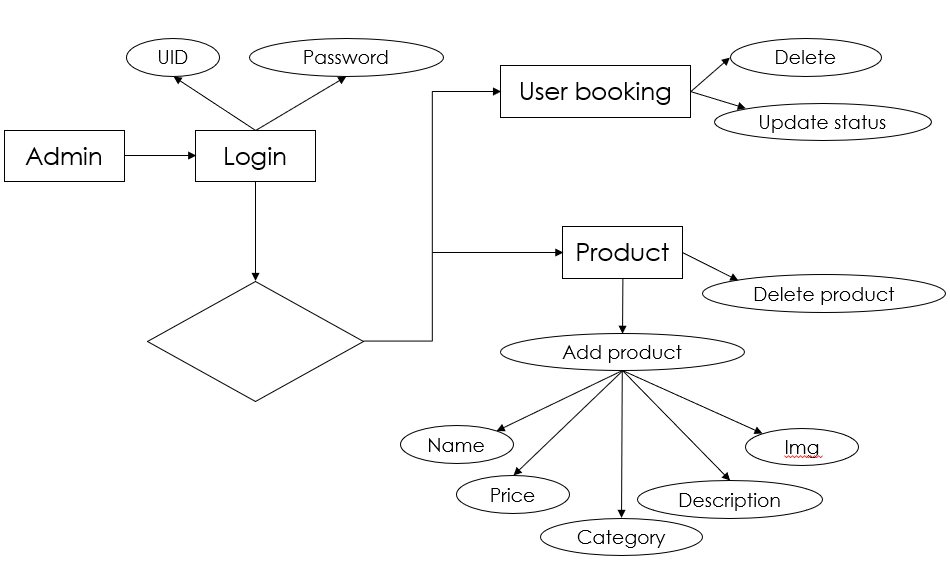
****

**Sequence Diagram (User)**

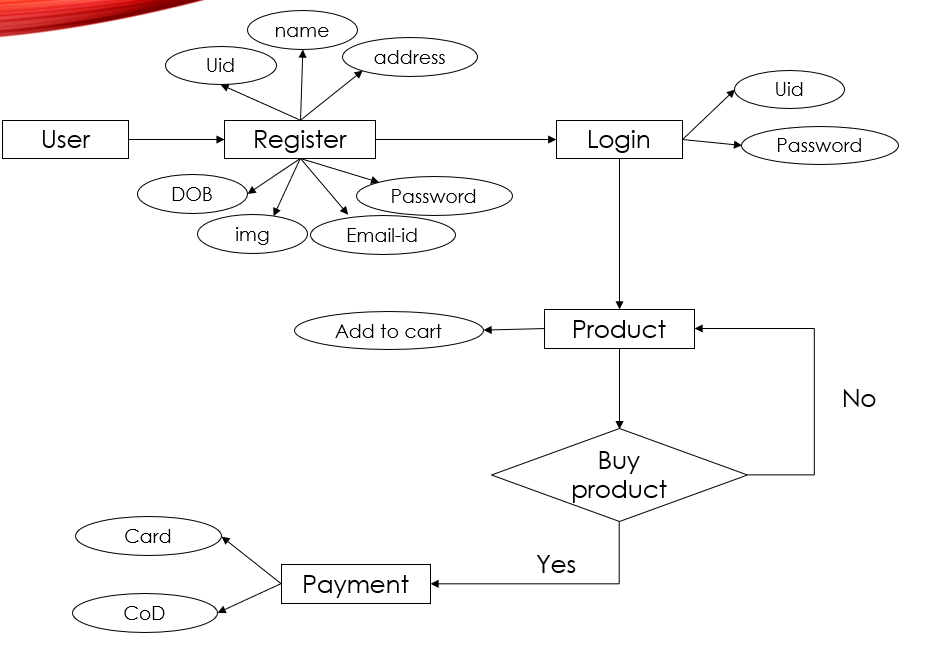
**Data Flow Diagram**



**Entity Relationship Diagrams (ER-Diagrams) (Admin):**

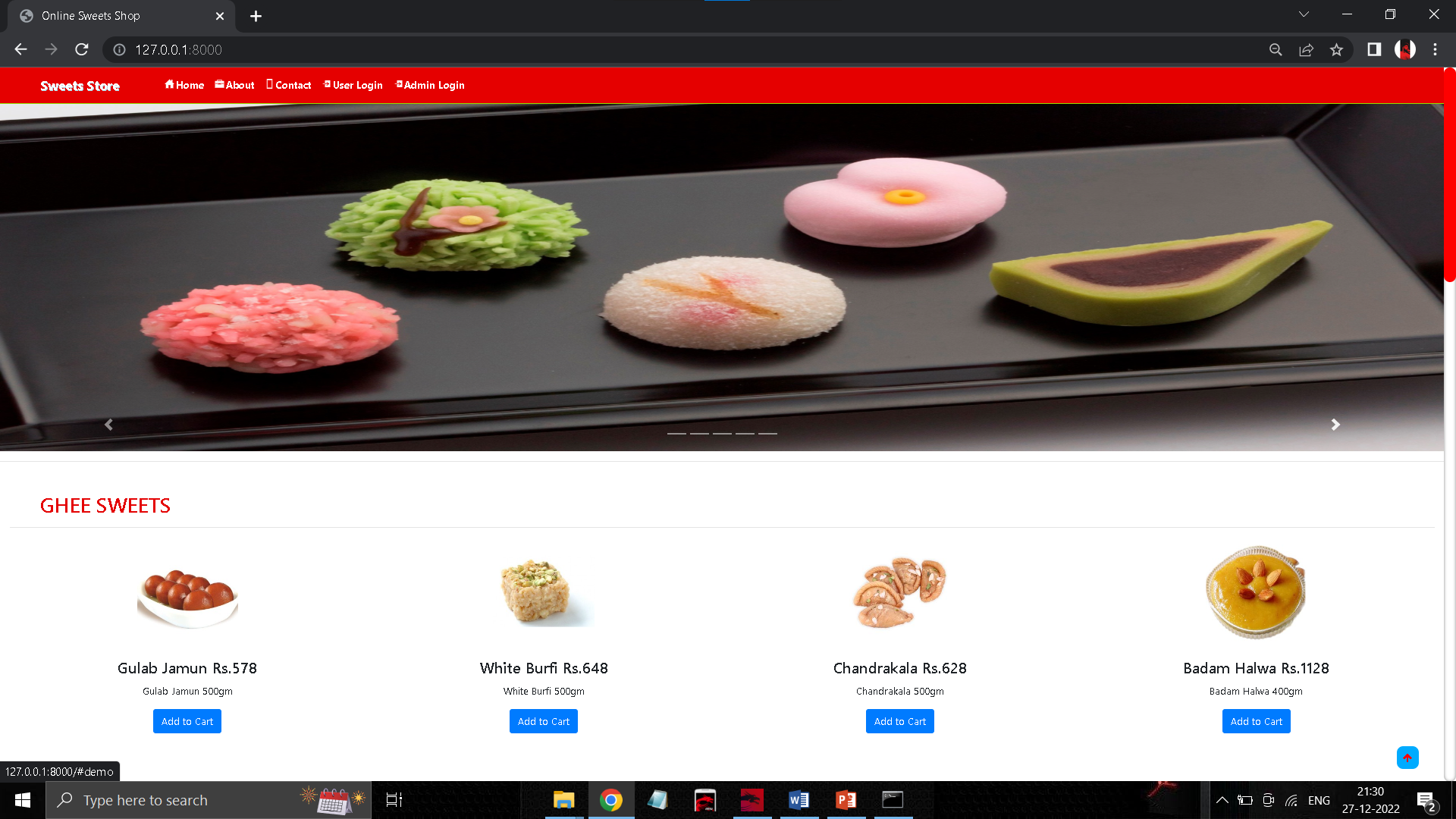
****

**Entity Relationship Diagrams (ER-Diagrams) (User):**

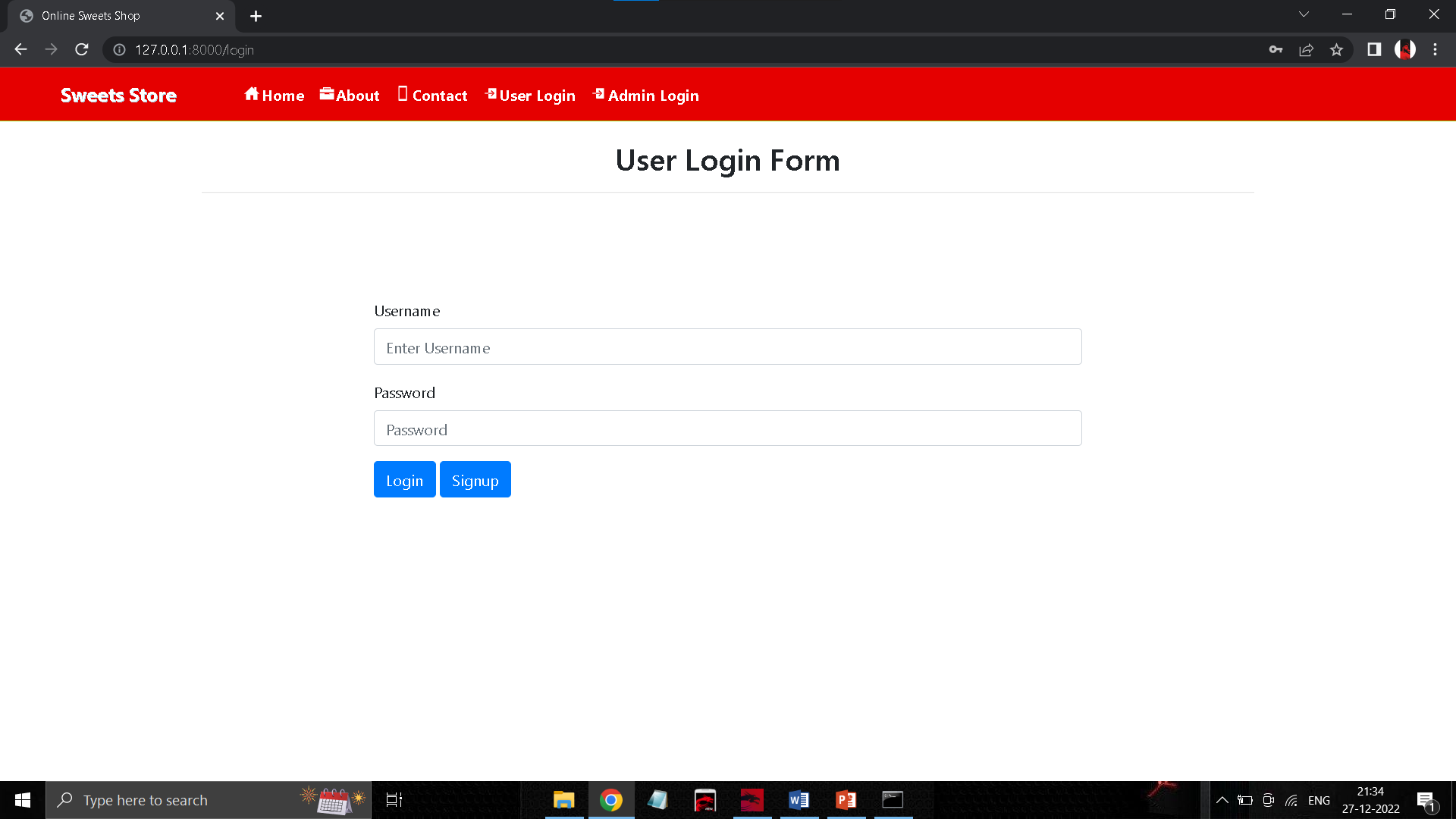
****

**Output screens**

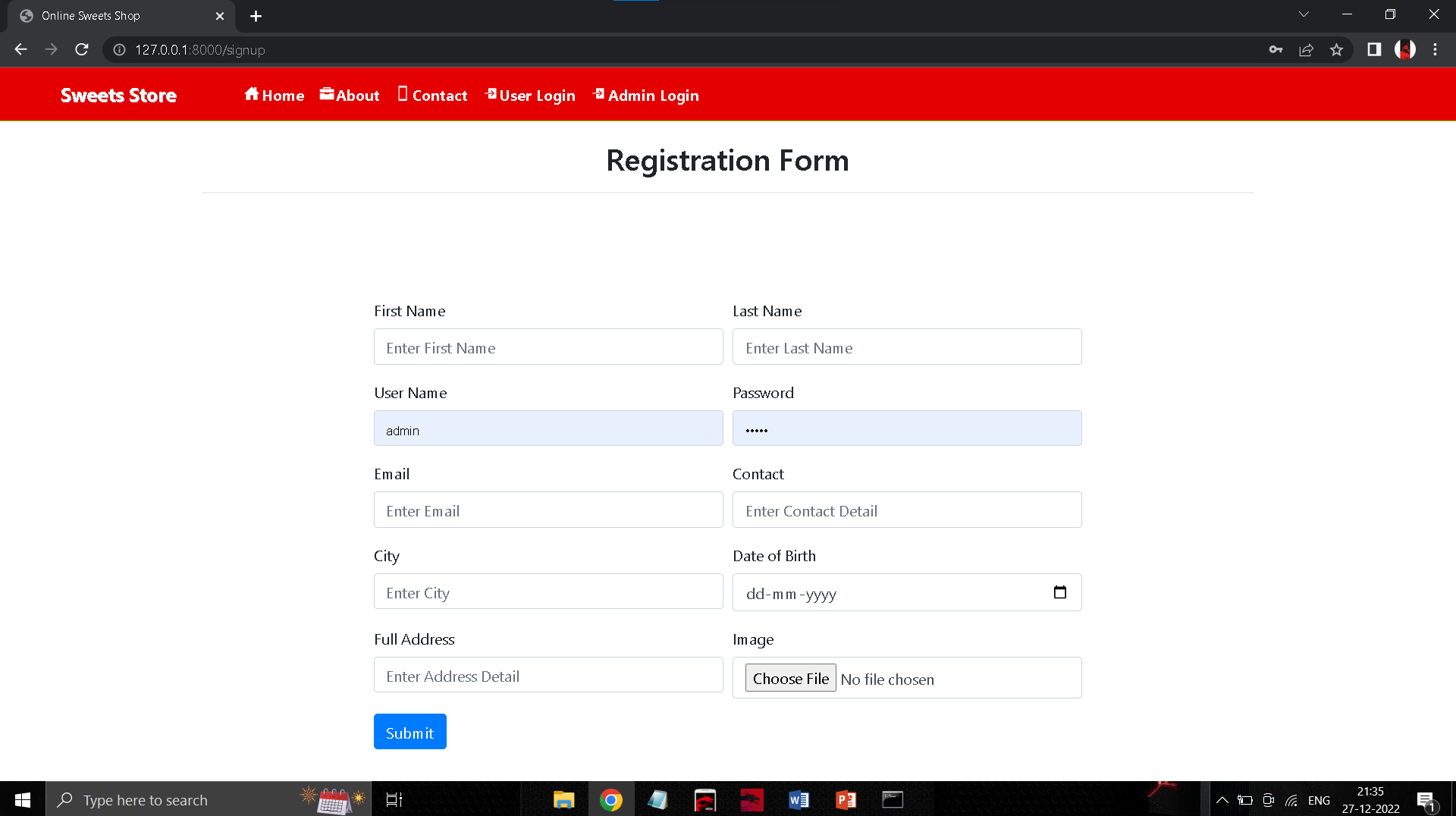
**HOME PAGE**



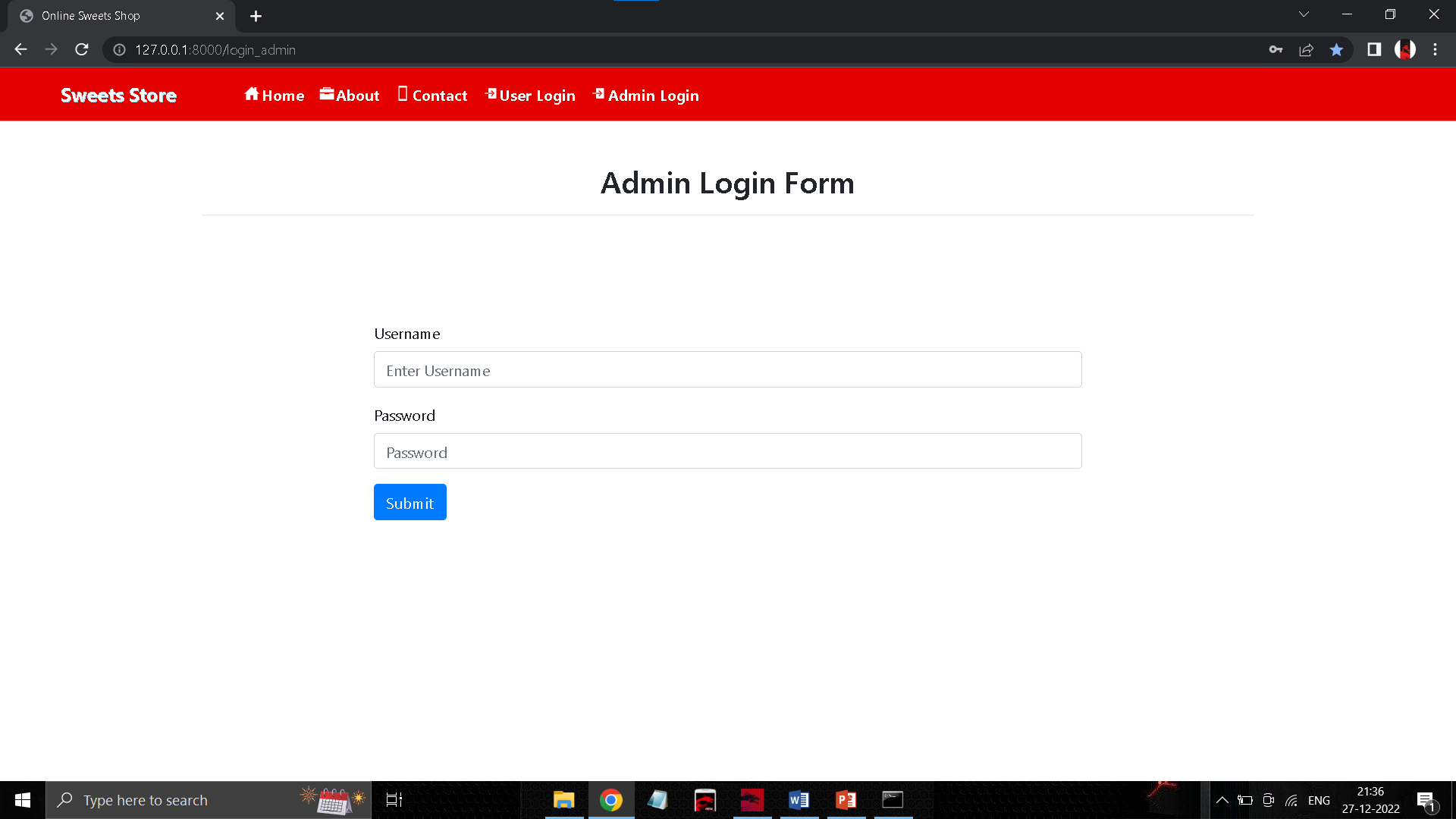
**User LOGIN PAGE**

****

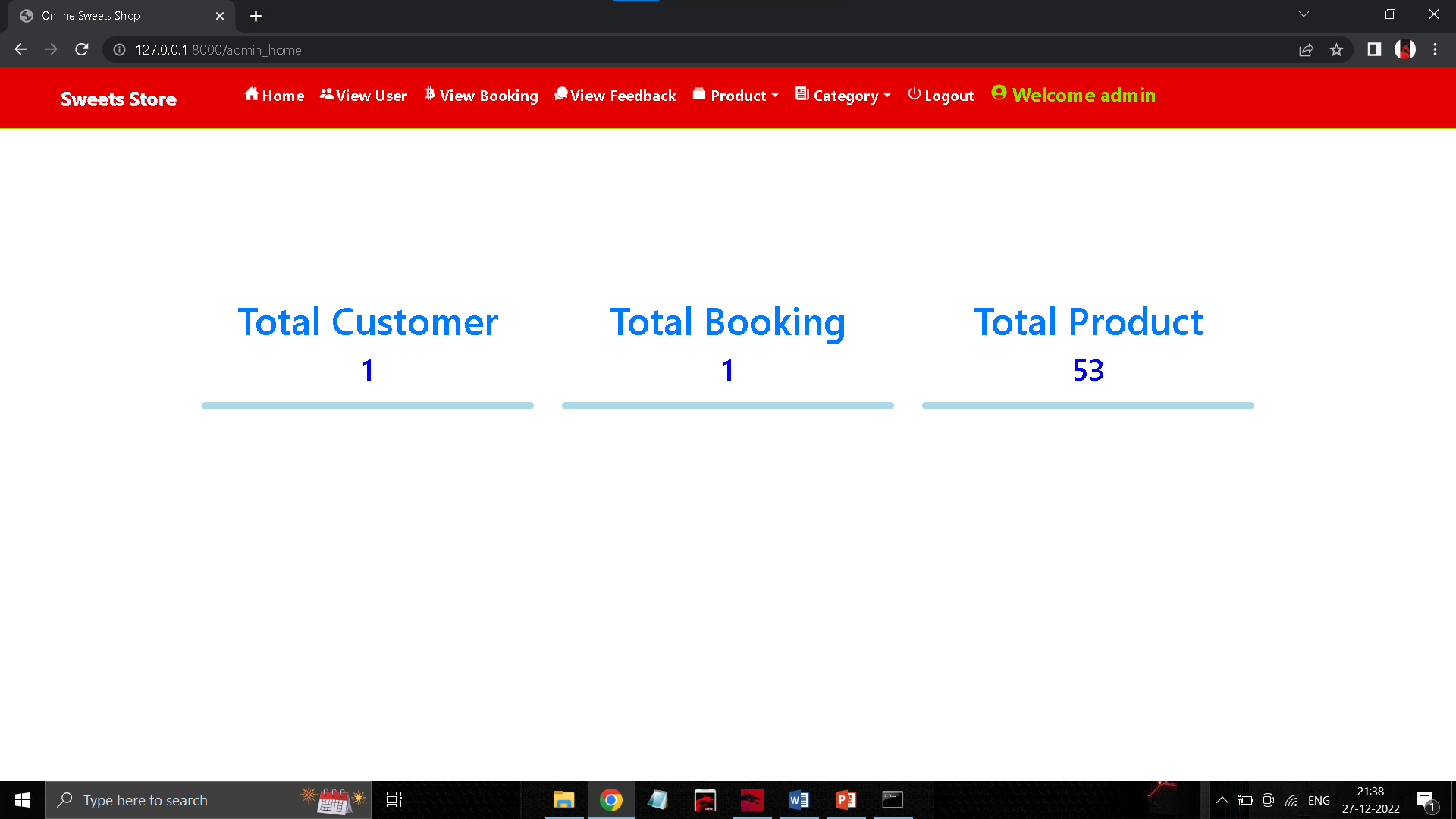
**SIGNUP PAGE**

****

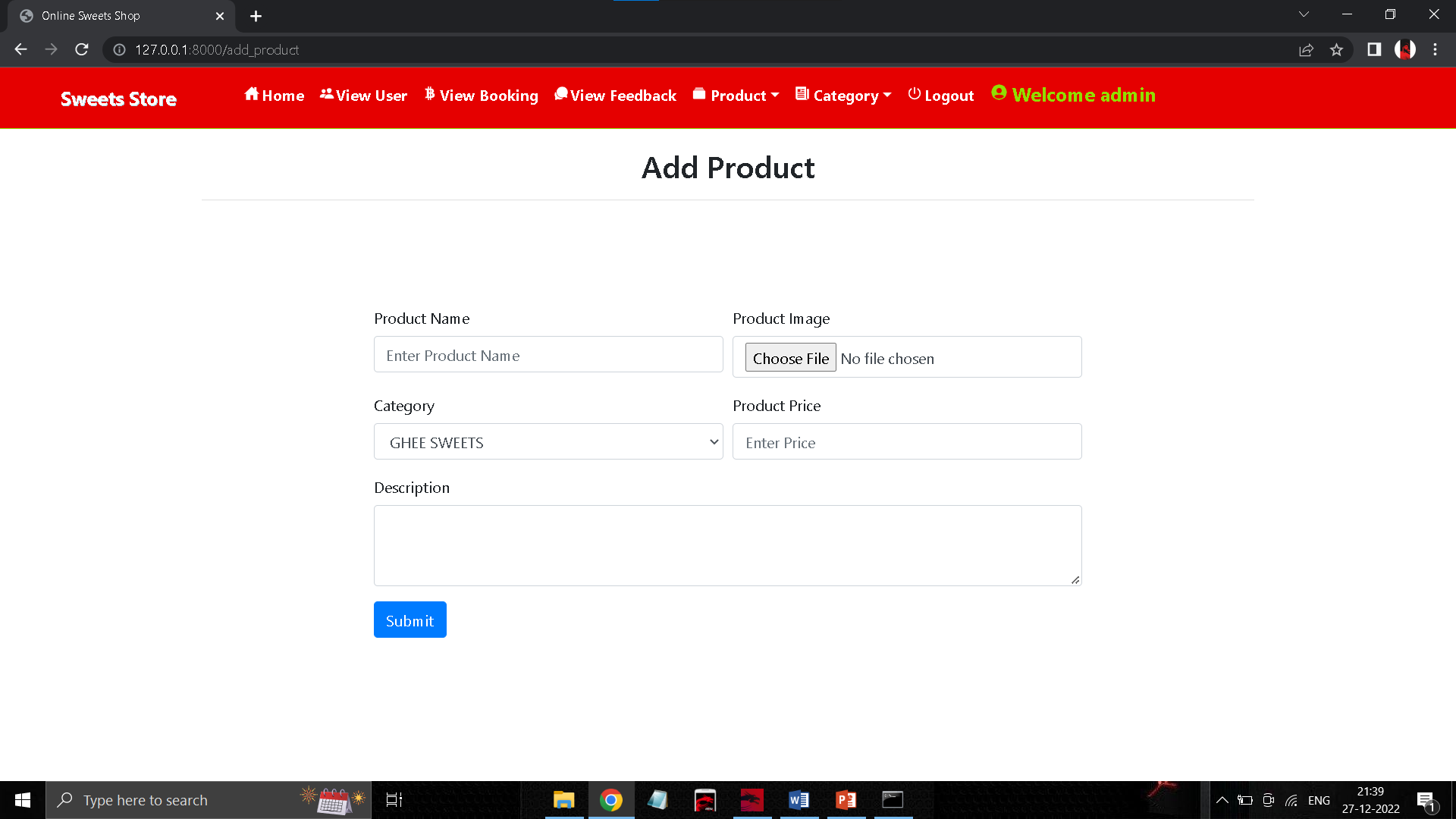
**ADMIN LOGIN PAGE**



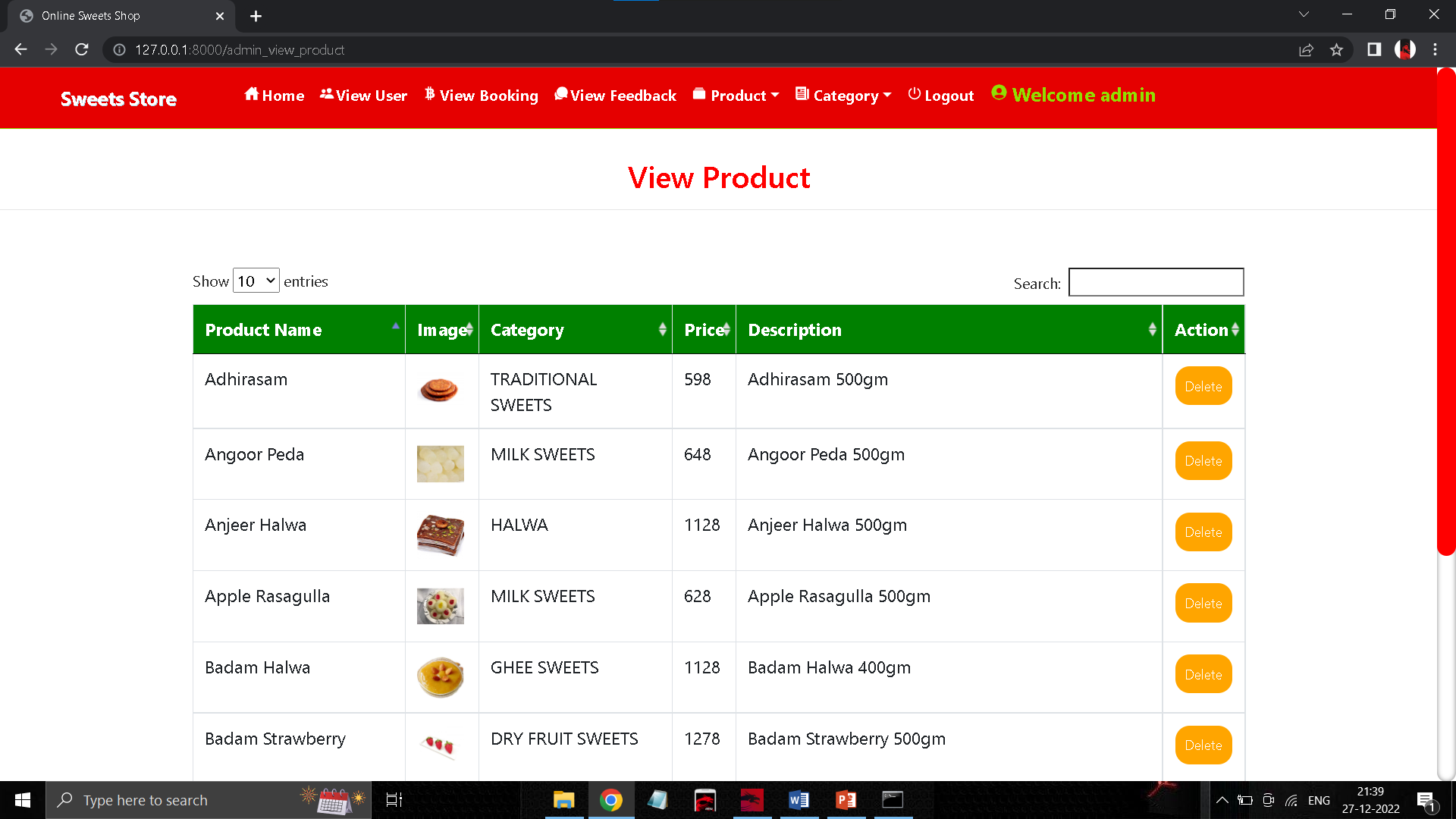
**ADMIN HOME PAGE**

****

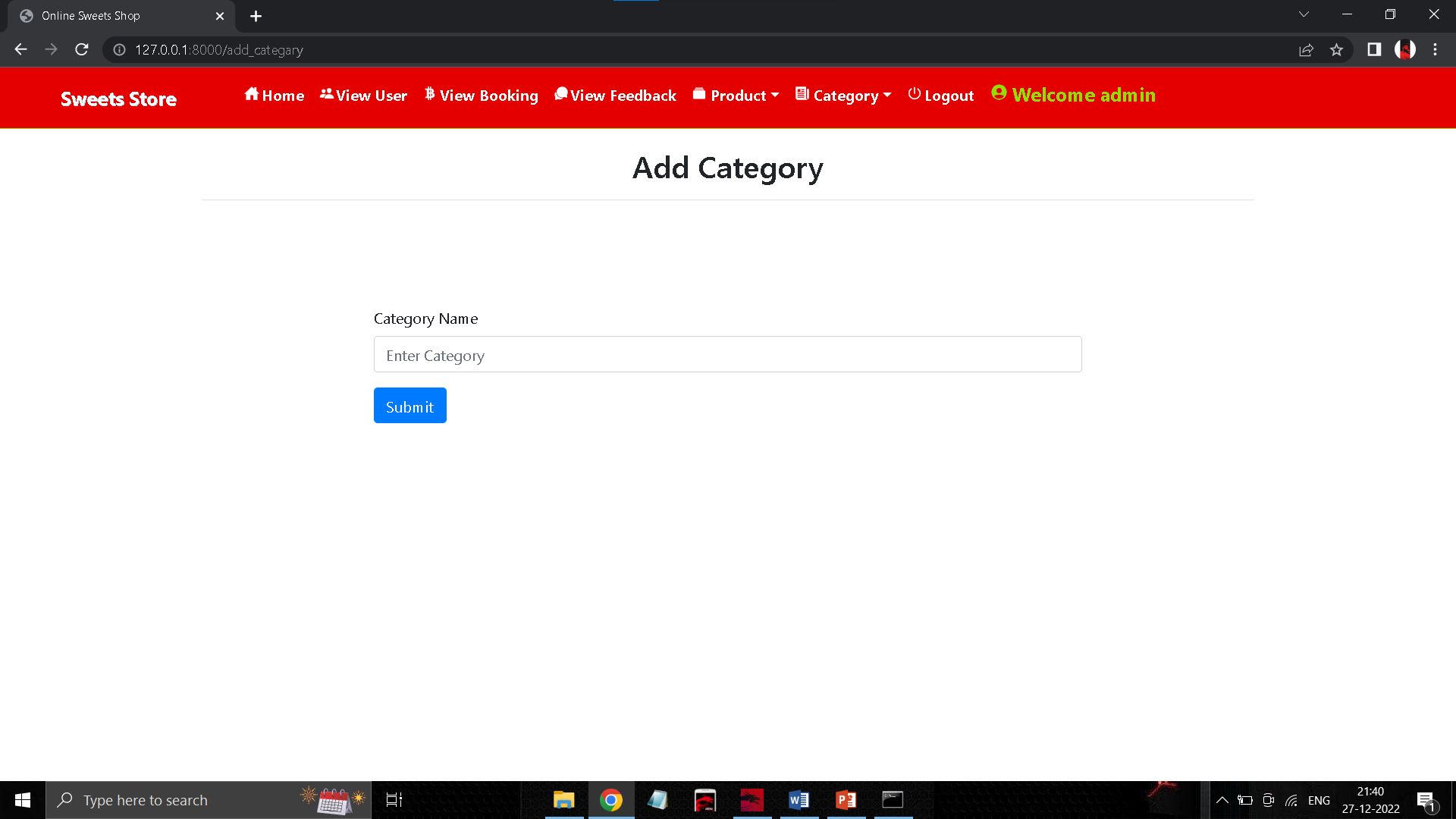
**ADD PRODUCT PAGE**



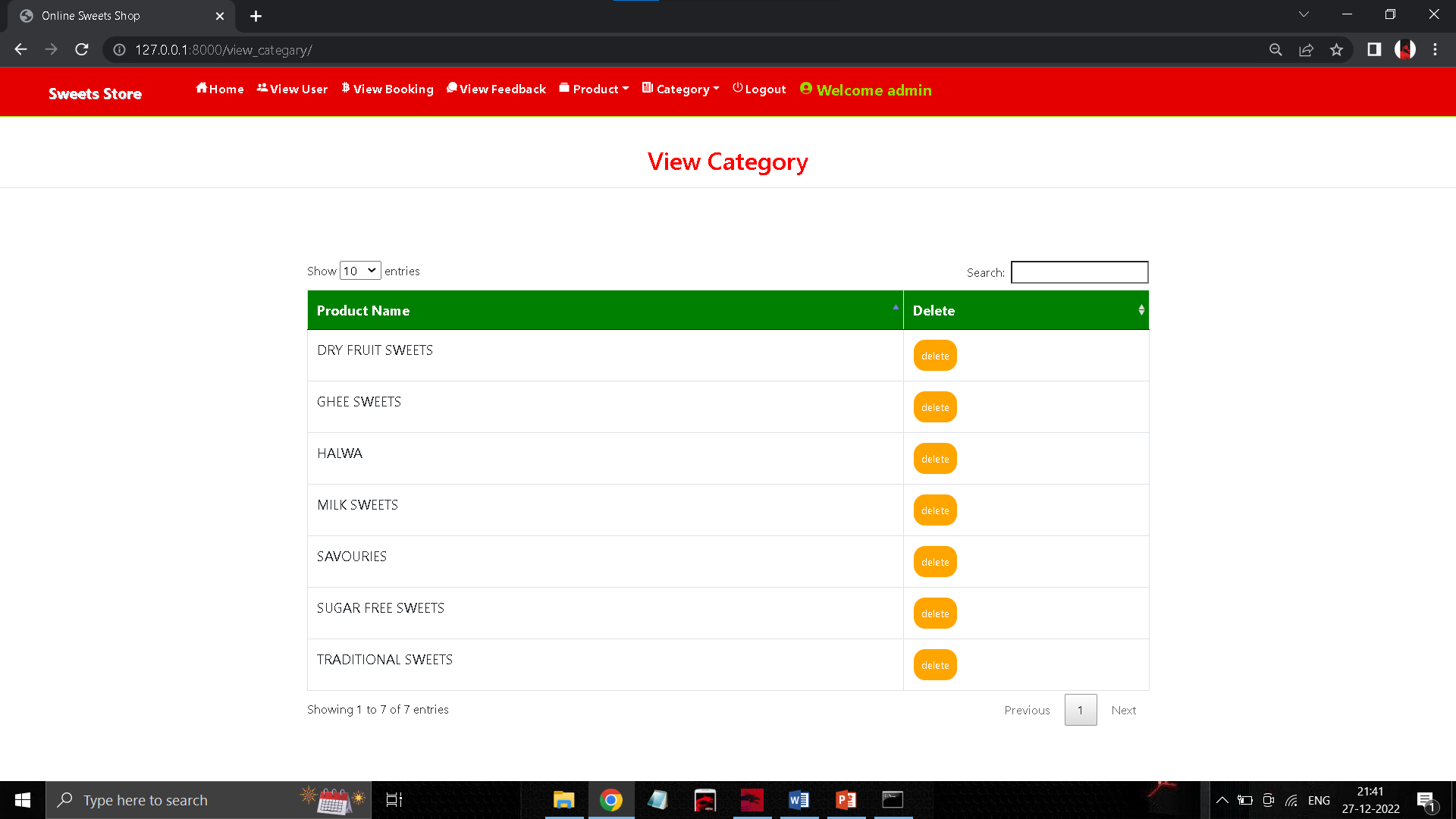
**VIEW PRODUCT PAGE**



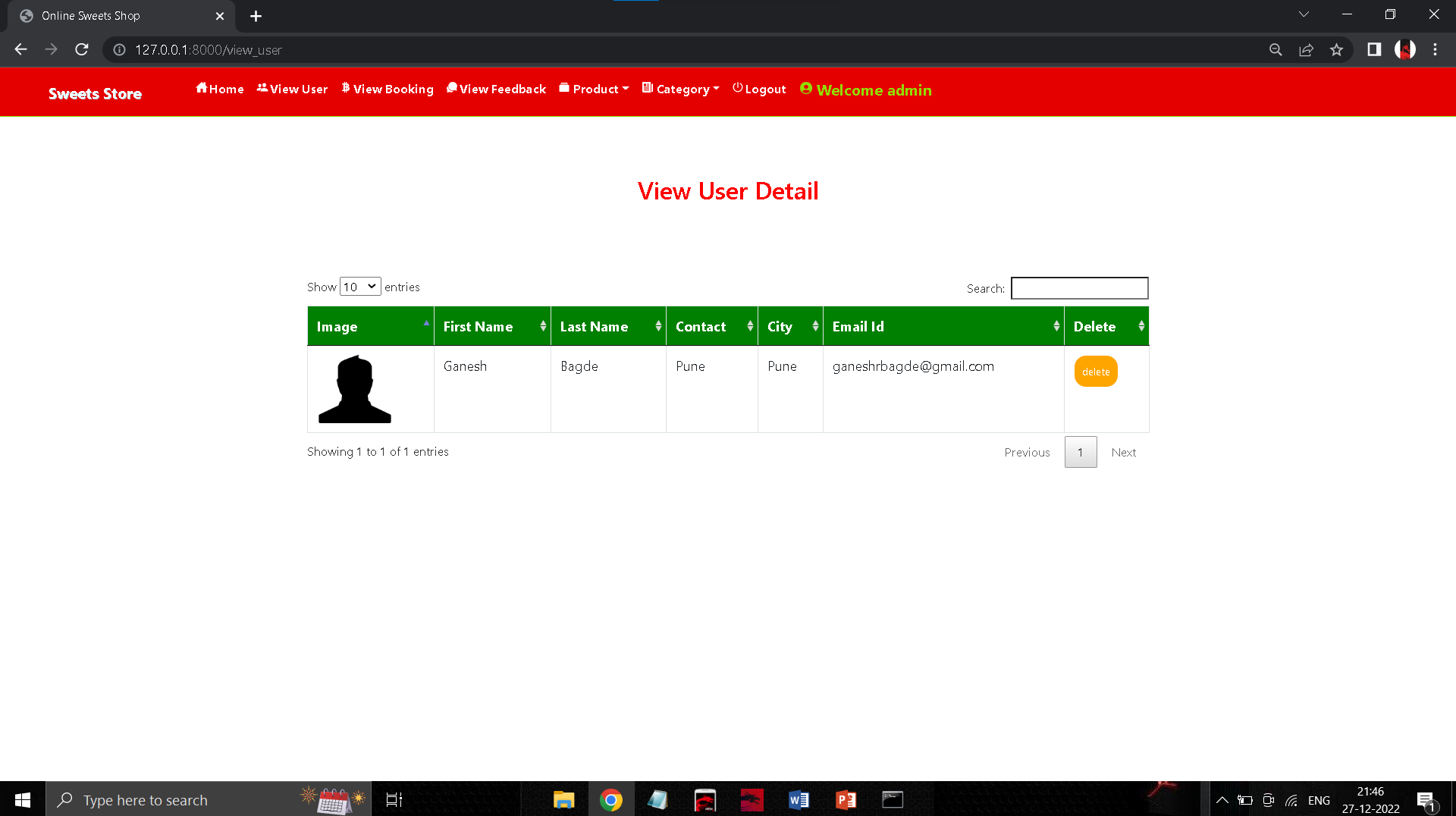
**ADD CATEGORY PAGE**



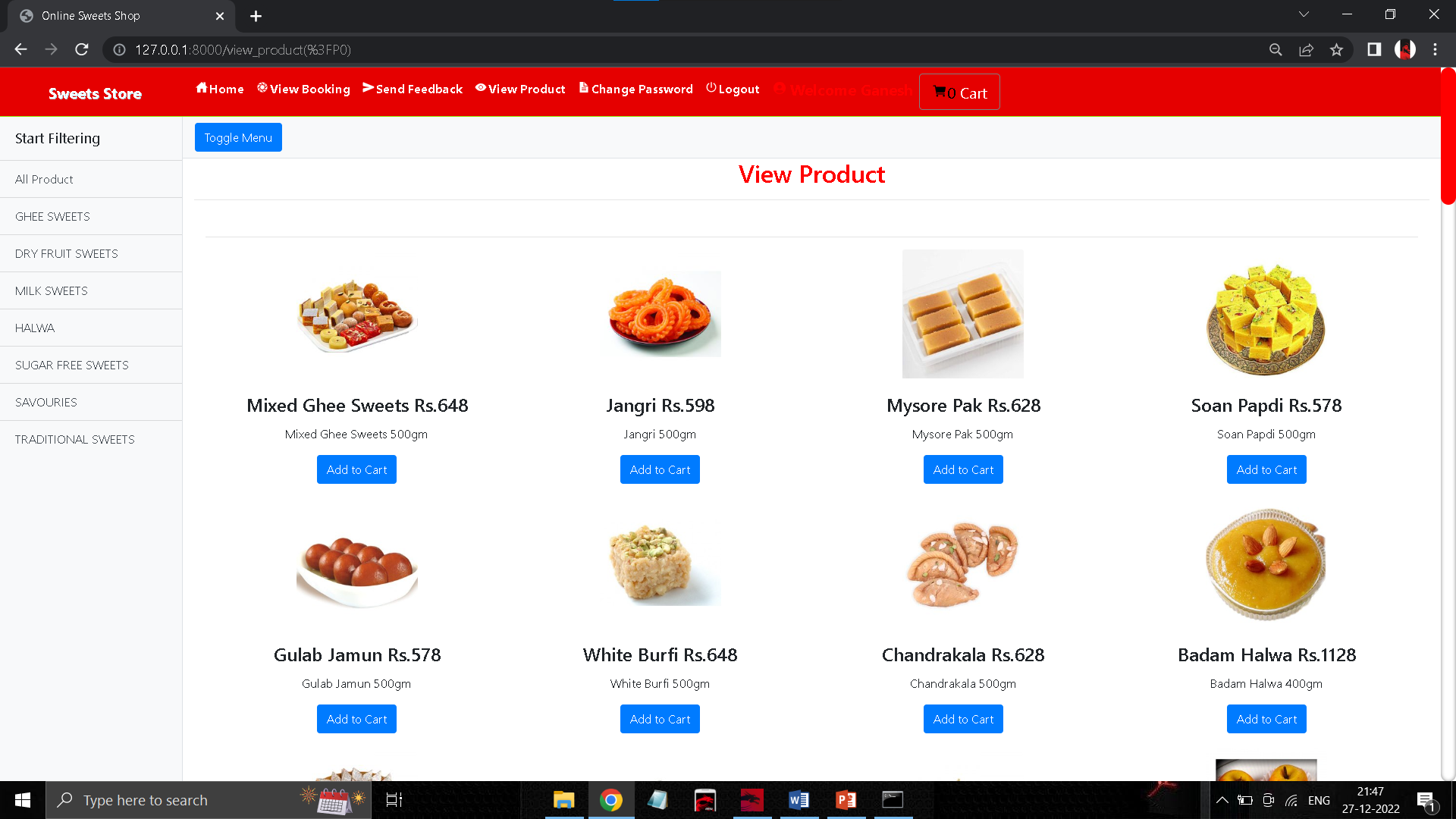
**VIEW CATEGORY PAGE**

****

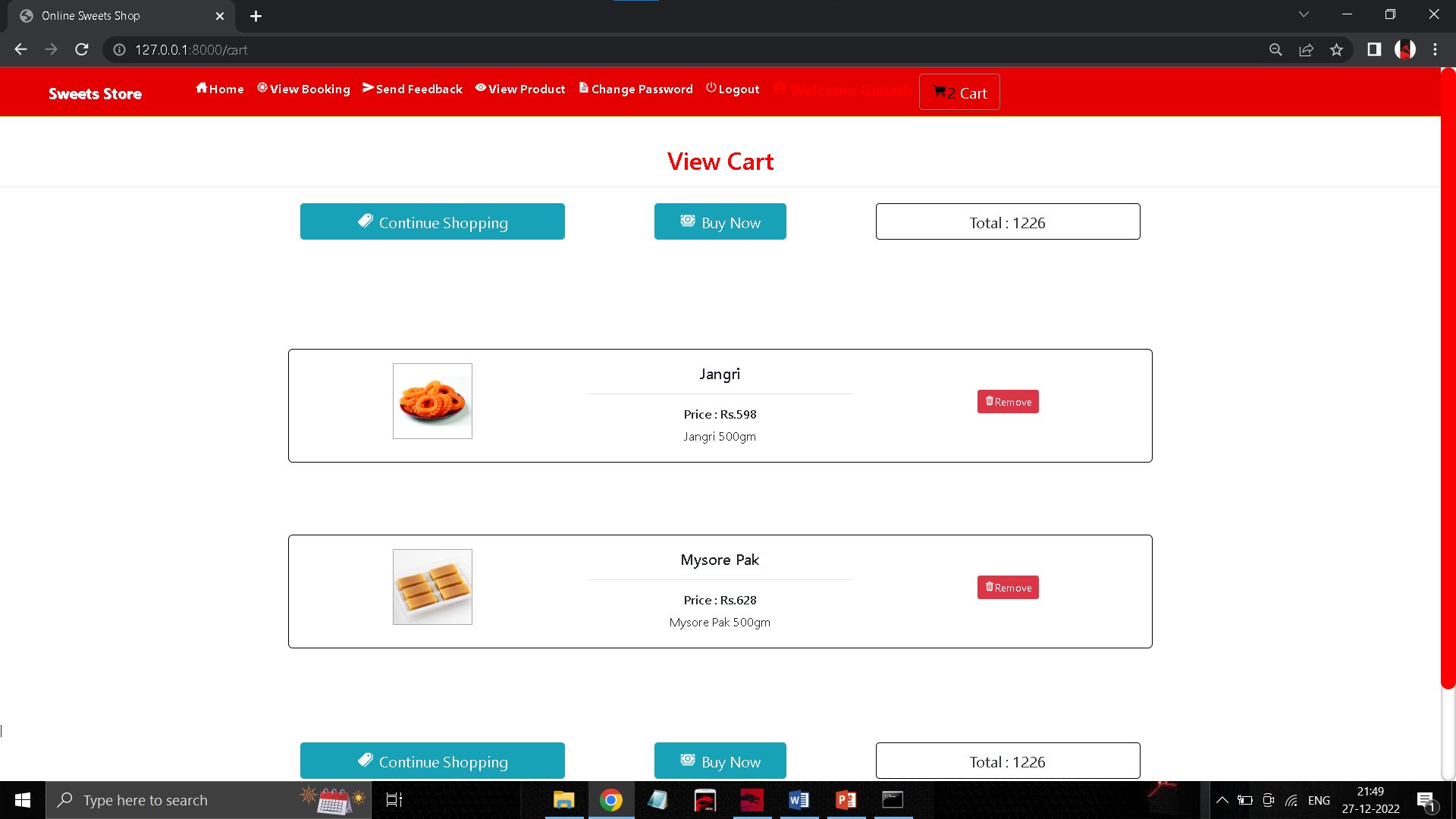
**VIEW USER DETAIL PAGE**

****

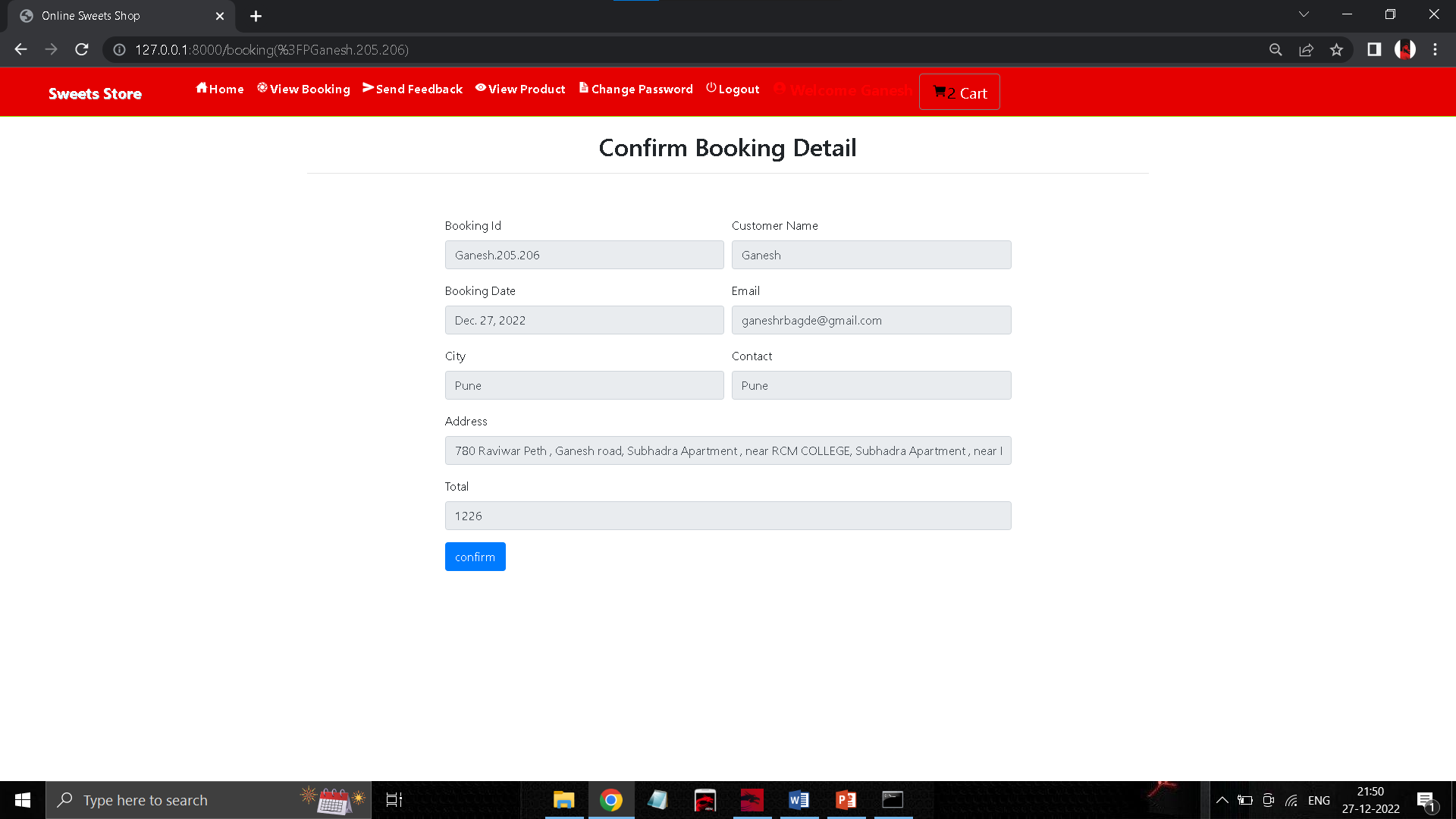
**VIEW PRODUCT PAGE (customer)**

****

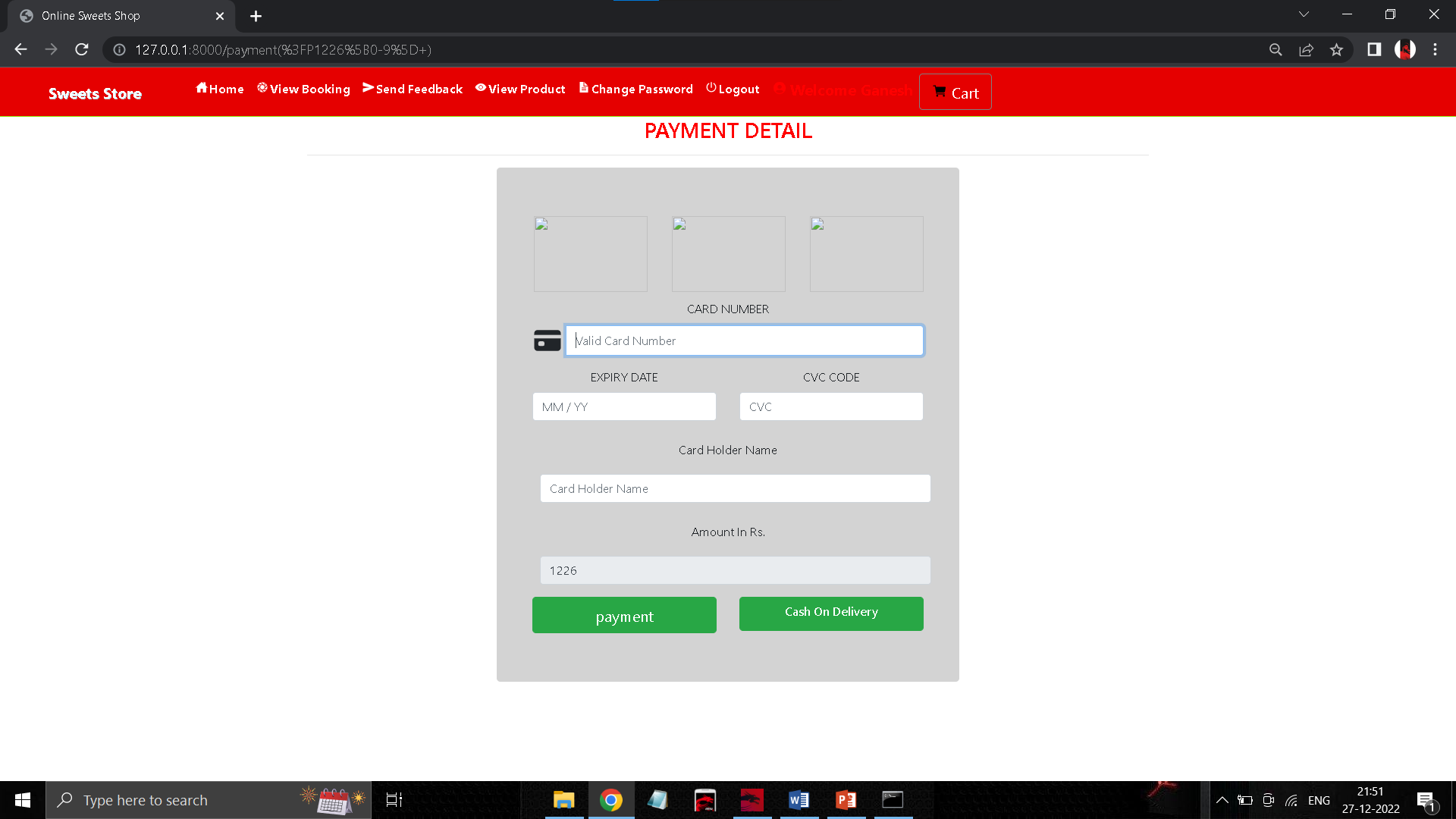
**VIEW CART PAGE**



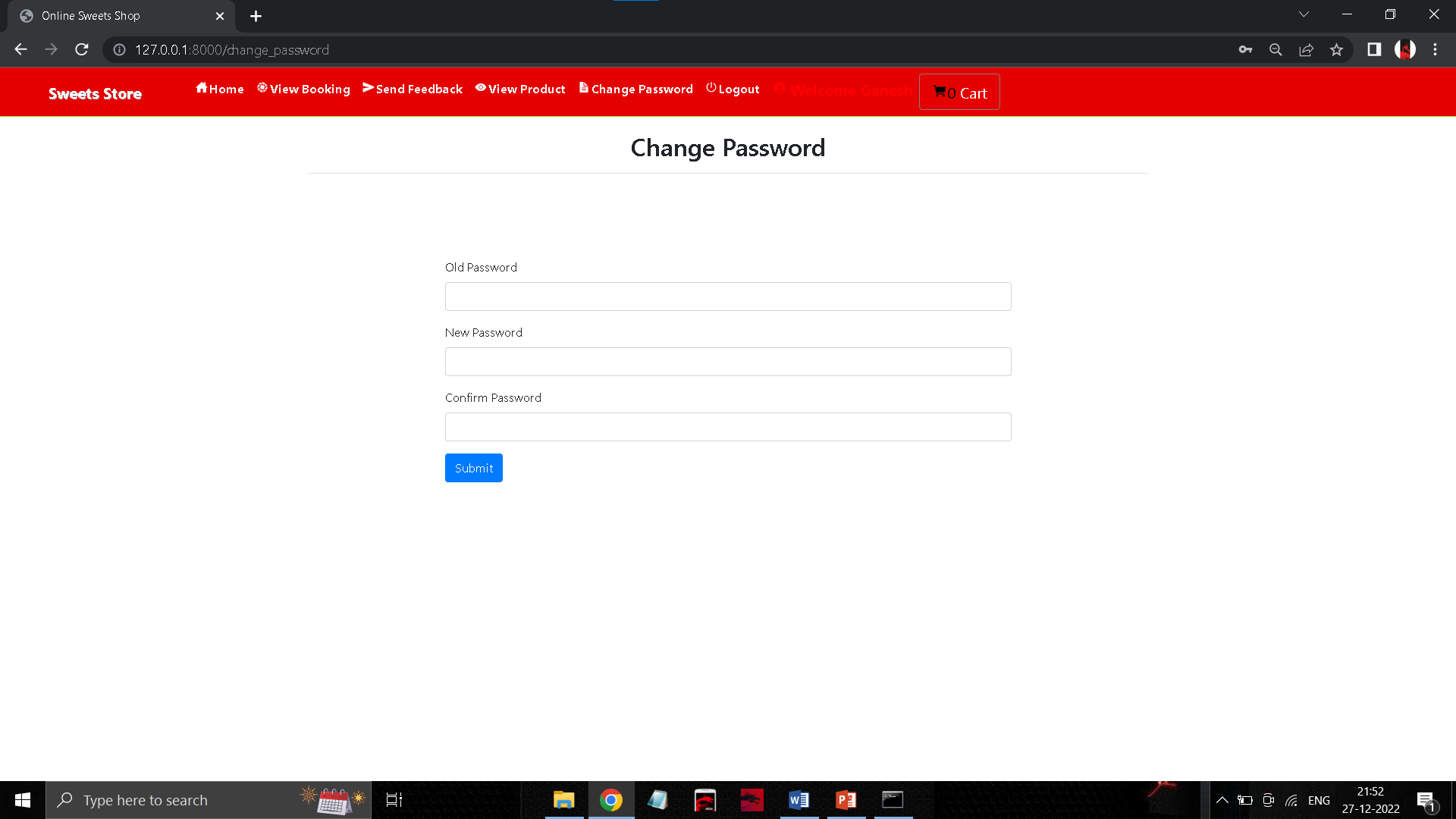
**CONFIRM BOOKING PAGE**

****

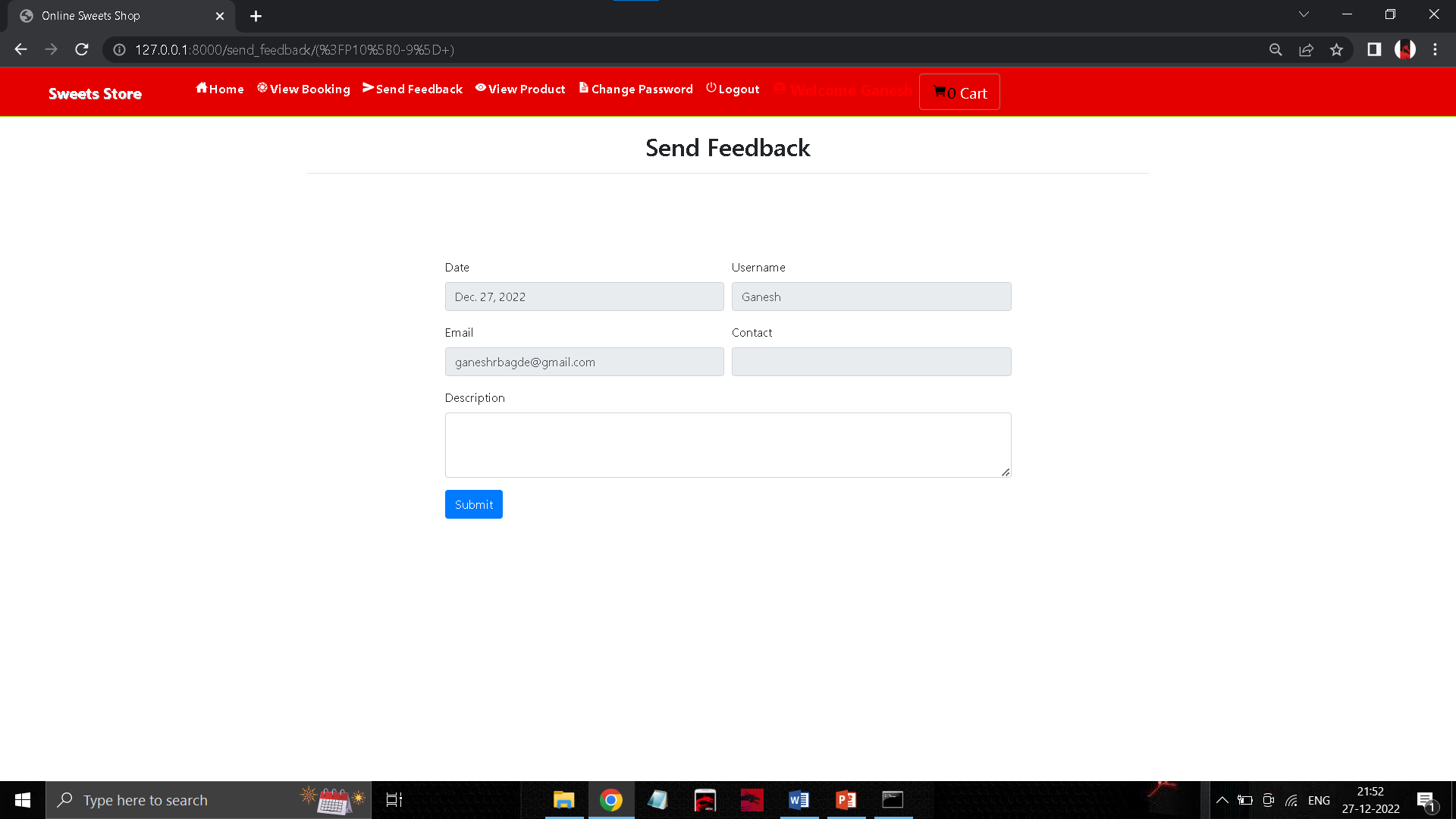
**PAYMENT DETAIL PAGE**



**CHANGE PASSWORD PAGE**



**SEND FEEDBACK PAGE**



**Testing Strategies**

Effective testing strategies ensure that the sweet shop management system is reliable, functional, and user-friendly. Here’s a comprehensive overview of the testing strategies to be employed:

* **Testing Tools:**

**- Unit Testing: JUnit (Java), PyTest (Python) for automated testing.**

* System Testing

- Scope: Test the complete system as a whole.

- Focus: Validate end-to-end functionalities such as processing sales, managing inventory, and generating reports.

- Environment: Mimic the production environment as closely as possible.

* Acceptance Testing

- Scope: Validate the system against user requirements.

- Tools: Cucumber for behaviour-driven development (BDD).

- Focus: Ensure the system meets the business needs and user expectations.

* Implementation Approach
* Test Planning: Define the objectives, resources, scope, and schedule of testing activities.
* Test Case Development: Create detailed test cases based on the system requirements and use cases.
* Environment Setup: Prepare the testing environment to closely resemble the production environment.
* Test Execution: Run the tests as per the test plan, recording the outcomes and any defects.
* Defect Tracking: Use tools like JIRA or Bugzilla to log and manage defects.
* Review and Feedback: Regularly review test results with the development team and stakeholders to ensure continuous improvement.

By employing these testing strategies, the Sweet Shop management system can be thoroughly vetted for functionality, performance, security, and user satisfaction, leading to a robust and reliable product.

**DRAWBACKS**

* Initial Cost and Complexity:

- High Implementation Cost: Setting up the system can be expensive, including hardware, software, and training expenses.

- Complexity: Implementing and maintaining the system requires technical expertise, which may be challenging for small businesses.

* Dependence on Technology:

- System Downtime: Reliance on technology means that system failures or downtimes can disrupt business operations.

- Technical Issues: Potential technical glitches or bugs may require ongoing support and maintenance.

* Data Security Concerns:

- Cybersecurity Risks: Storing sensitive customer and business data online increases the risk of cyberattacks and data breaches.

* Scalability and Flexibility:

- Scalability Issues: The system may require significant upgrades or replacements as the business grows.

- Customization Limitations: Off-the-shelf solutions might not fully meet the unique needs of the sweet shop, requiring custom development.

These drawbacks need to be carefully considered and mitigated through strategic planning, investment in training, robust cybersecurity measures, and choosing scalable solutions.

**CONCLUSION**

In conclusion, while the computerized sweet shop management system offers numerous benefits such as improved efficiency, enhanced customer experience, and streamlined operations, it also presents several drawbacks. These include dependency on technology, the need for extensive training, initial investment and maintenance costs, security risks, and potential resistance to change among employees. Despite these challenges, with careful planning, implementation, and ongoing management, the system can significantly enhance the overall effectiveness and competitiveness of the sweet shop business.

**REFERENCE**

* <https://docs.python.org/3/tutorial/index.html>
* <https://docs.djangoproject.com/en/5.0/>
* <https://www.w3schools.com/python/>
* <https://www.w3schools.com/django/>
* <https://www.youtube.com/watch?v=_uQrJ0TkZlc>

THANK YOU