Creating an application based on APEX technology

Objective of the Study: Development of practical skills in creating applications based on APEX technology by engaging in hands-on design and implementation of database structures, user interfaces, and business logic. This includes mastering the process of building relational tables with appropriate constraints, developing views and functions for efficient data manipulation, and creating interactive web pages that support CRUD operations and data visualization. Additionally, the project aims to deepen understanding of user authentication mechanisms and how to manage different access levels within an APEX application, thereby preparing for real-world application development scenarios.

Tasks:

1. Create five tables:
    - The tables must be related to each other with a one-to-many (1-M) relationship.
2. Develop three views and three functions:
    - Views should provide efficient data representation.
    - Functions should perform data processing and manipulation tasks.
3. Create three pages with CRUD functionality:
    - Enable adding, editing, and deleting records in the tables.
4. Develop pages to display data from the views:
    - Provide options for filtering and sorting the displayed data.
5. Create pages with and without authentication:
    - Pages with authentication should allow authorized users to edit data.
    - Pages without authentication are intended for external users to view data only.

Conducted by Maksym Alieksieiev

October 2024

Workflow

1. **Database Schema Design and Table Relationships**

In the APEX environment, five tables were created: Categories, Customers, Orders, Products, and Reviews. Each table was assigned a Primary Key constraint to ensure the uniqueness of records and maintain data integrity within the database.

To establish relationships between these tables, Foreign Key constraints were implemented. These constraints enforce referential integrity by defining how records in one table relate to records in another.

Specifically, the following relationships were established:

- Orders –> Customers (1:N)
- Orders –> Products (1:N)
- Reviews –> Customers (1:N)
- Reviews –> Products (1:N)
- Products –> Categories (1:N)

2. **Data Population and Resulting Dataset**

After defining the database schema and establishing relationships between the tables, corresponding data were inserted into each of the tables. This step involved populating the tables with sample data relevant to the application domain.

The data insertion process ensured that all foreign key constraints were respected, meaning that every reference between tables correctly corresponded to existing records.

As a result of this data population, the database contained a consistent and interconnected dataset that could be used for application testing, querying, and reporting. The following dataset snapshot or query results illustrate the current state of the database after the data was added:

CATEGORIES

Columns | Data | Indexes | Constraints | Grants | Statistics | Triggers | Dependencies | DDL | Sample Queries

+ Insert Row | Columns... | Filter... | Count Rows | Load Data | Download | Refresh

CATEGORY_ID	CATEGORY_NAME	CATEGORY_DESCRIPTION
7	Beauty	includes cosmetics, skincare products, haircare items, and other beauty-related products ...
4	Sports	features products associated with sports and physical activities, including sports equipm...
5	Home&Garden	consists of products related to home improvement, decor, gardening tools, furniture, and...
1	Electronics	includes electronic devices such as smartphones, laptops, cameras, smartwatches, and o...
2	Accessories	encompasses additional items that complement electronic devices, such as headphones, ...
3	Clothing	includes various types of apparel, including shirts, pants, dresses, and other garments de...
6	Books	encompasses written works, including novels, non-fiction, educational materials, and oth...

Figure 1 – Table “Categories”

CUSTOMERS

Columns | Data | Indexes | Constraints | Grants | Statistics | Triggers | Dependencies | DDL | Sample Queries

+ Insert Row | Columns... | Filter... | Count Rows | Load Data | Download | Refresh

CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	ADDRESS	CITY
1	John	Doe	john.doe@email.com	123-456-7890	123 Main St, Apt 101	New York
2	Jane	Smith	jane.smith@email.com	987-654-3210	456 Oak St, Suite 202	Los Angeles
3	Alex	Johnson	alex.j@email.com	555-123-4567	789 Pine Rd, #303	Chicago
41	Patrick	Smith	patrick.smith@gmail.com	121-323-1222	293 Main avenue, 209	Dallas
5	Robert	White	robert.white@gmail.com	777-888-0000	890 Cedar Blvd, #505	San Francisco
6	Emily	Brown	emily.brown@email.com	222-333-4444	111 Willow Ln, Apt 606	Seattle
7	Michael	Davis	michael.d@email.com	666-777-8888	222 Birch St, #707	Austin
4	Maria	Rodriguez	maria.rodr@email.com	333-999-1111	567 Elm Ave, Unit 404	Miami

Figure 2 – Table “Customers”

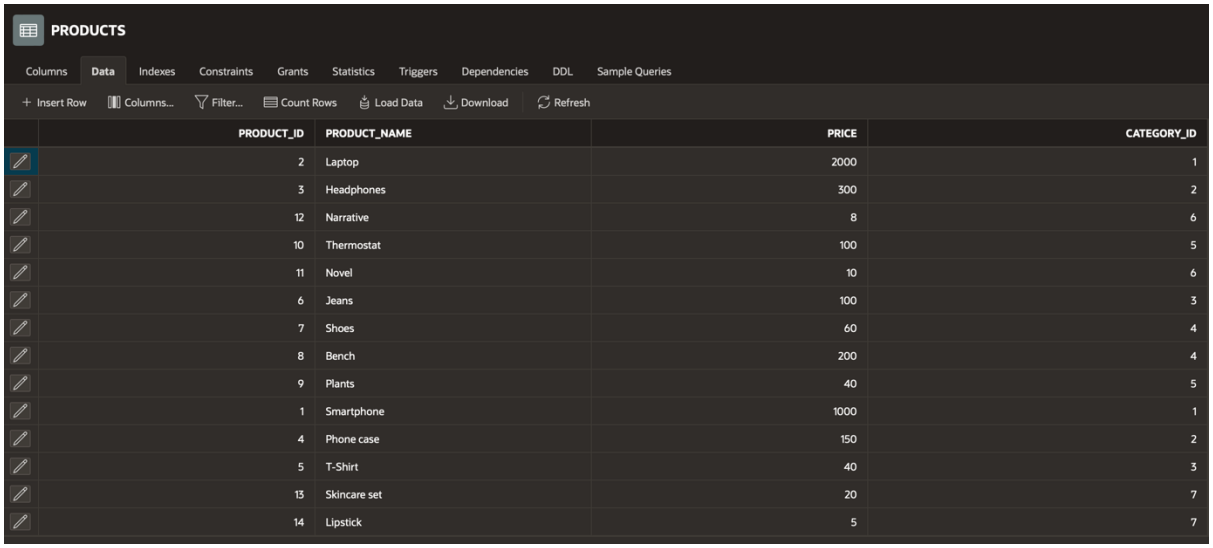
ORDERS

Columns | Data | Indexes | Constraints | Grants | Statistics | Triggers | Dependencies | DDL | Sample Queries

+ Insert Row | Columns... | Filter... | Count Rows | Load Data | Download | Refresh

ORDER_ID	CUSTOMER_ID	ORDER_DATE	QUANTITY	TOTAL_AMOUNT	PRODUCT_ID
2	2	01/01/2023	1	300	3
7	7	03/01/2023	2	120	7
4	4	02/01/2023	1	100	10
5	5	02/01/2023	4	40	11
6	6	02/01/2023	2	200	6
1	1	01/01/2023	2	4000	2
3	3	01/01/2023	3	24	12

Figure 3 – Table “Orders”



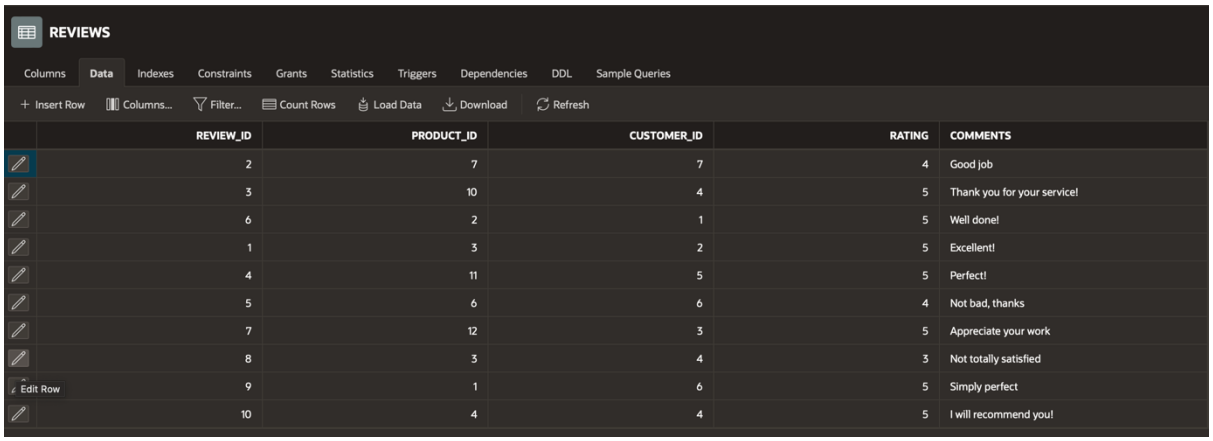
PRODUCTS

Columns | Data | Indexes | Constraints | Grants | Statistics | Triggers | Dependencies | DDL | Sample Queries

+ Insert Row | Columns... | Filter... | Count Rows | Load Data | Download | Refresh

PRODUCT_ID	PRODUCT_NAME	PRICE	CATEGORY_ID
2	Laptop	2000	1
3	Headphones	300	2
12	Narrative	8	6
10	Thermostat	100	5
11	Novel	10	6
6	Jeans	100	3
7	Shoes	60	4
8	Bench	200	4
9	Plants	40	5
1	Smartphone	1000	1
4	Phone case	150	2
5	T-Shirt	40	3
13	Skincare set	20	7
14	Lipstick	5	7

Figure 4 – Table “Products”



REVIEWS

Columns | Data | Indexes | Constraints | Grants | Statistics | Triggers | Dependencies | DDL | Sample Queries

+ Insert Row | Columns... | Filter... | Count Rows | Load Data | Download | Refresh

REVIEW_ID	PRODUCT_ID	CUSTOMER_ID	RATING	COMMENTS
2	7	7	4	Good job
3	10	4	5	Thank you for your service!
6	2	1	5	Well done!
1	3	2	5	Excellent!
4	11	5	5	Perfect!
5	6	6	4	Not bad, thanks
7	12	3	5	Appreciate your work
8	3	4	3	Not totally satisfied
9	1	6	5	Simply perfect
10	4	4	5	I will recommend you!

Edit Row

Figure 5 – “Table “Reviews”

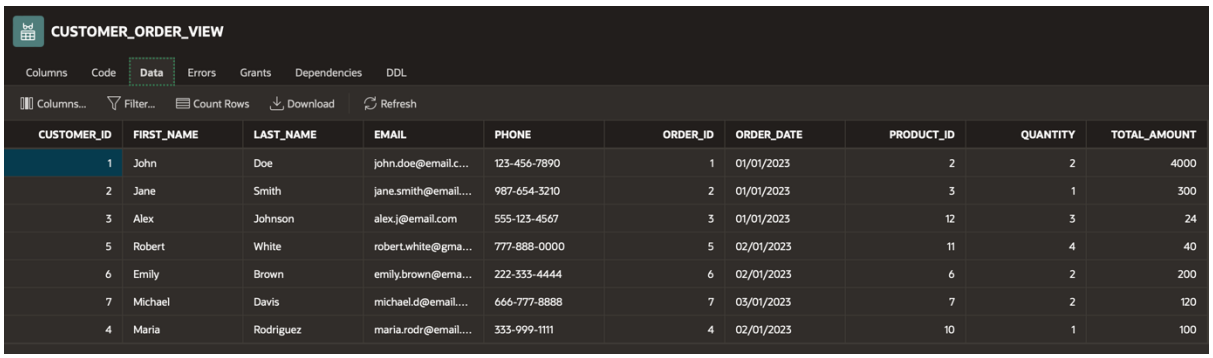
3. Development of Database Views

To facilitate easier access and improved data retrieval for common queries, three database views were developed:

- **Customer_Order_View** – provides a comprehensive overview of each customer’s orders. This view simplifies querying order history by combining customer details with their corresponding orders in a single, readable dataset.

- **Product_Category_View** – allows quick access to product details along with their associated category information. This view is particularly useful for filtering or grouping products based on their categories without needing to join tables manually in every query.
- **Review_Customer_View** – enables retrieval of customer reviews along with relevant customer information. This view supports analysis of customer feedback and can be used for reporting or monitoring product satisfaction.

These views enhance the application's ability to perform complex queries efficiently while maintaining data integrity and security by controlling direct access to underlying tables.


CUSTOMER_ORDER_VIEW

Columns | Code | Data | Errors | Grants | Dependencies | DDL

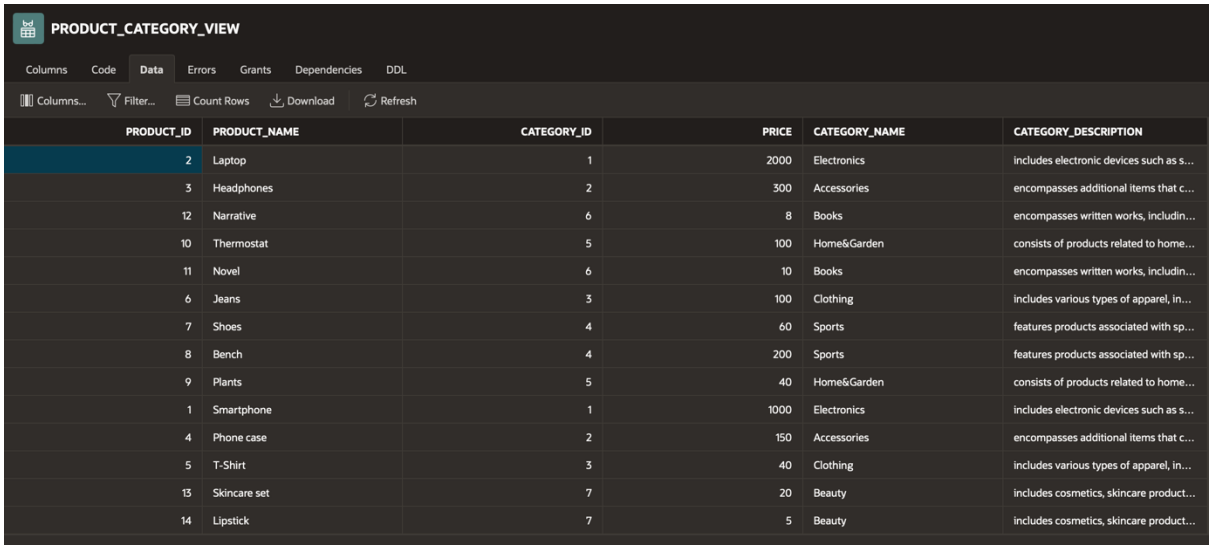
Columns... | Filter... | Count Rows | Download | Refresh

CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	ORDER_ID	ORDER_DATE	PRODUCT_ID	QUANTITY	TOTAL_AMOUNT
1	John	Doe	john.doe@email.c...	123-456-7890	1	01/01/2023	2	2	4000
2	Jane	Smith	jane.smith@email....	987-654-3210	2	01/01/2023	3	1	300
3	Alex	Johnson	alex.j@email.com	555-123-4567	3	01/01/2023	12	3	24
5	Robert	White	robert.white@gma...	777-888-0000	5	02/01/2023	11	4	40
6	Emily	Brown	emily.brown@ema...	222-333-4444	6	02/01/2023	6	2	200
7	Michael	Davis	michael.d@email....	666-777-8888	7	03/01/2023	7	2	120
4	Maria	Rodriguez	maria.rodr@email....	333-999-1111	4	02/01/2023	10	1	100

Figure 6 – View “Customer_Order_View”

SQL code:

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "CUSTOMER_ORDER_VIEW" ("CUSTOMER_ID", "FIRST_NAME", "LAST_NAME", "EMAIL", "PHONE", "ORDER_ID", "ORDER_DATE", "PRODUCT_ID", "QUANTITY", "TOTAL_AMOUNT") AS
SELECT c.customer_id, c.first_name, c.last_name, c.email, c.phone, o.order_id, o.order_date, o.product_id, o.quantity, o.total_amount
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id;
```



PRODUCT_CATEGORY_VIEW

Columns | Code | Data | Errors | Grants | Dependencies | DDL

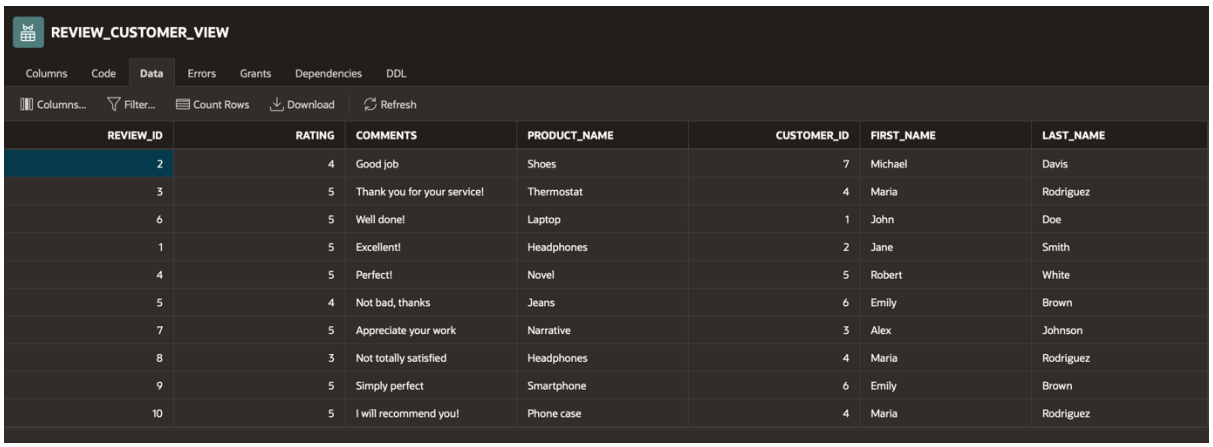
Columns... | Filter... | Count Rows | Download | Refresh

PRODUCT_ID	PRODUCT_NAME	CATEGORY_ID	PRICE	CATEGORY_NAME	CATEGORY_DESCRIPTION
2	Laptop	1	2000	Electronics	includes electronic devices such as s...
3	Headphones	2	300	Accessories	encompasses additional items that c...
12	Narrative	6	8	Books	encompasses written works, includin...
10	Thermostat	5	100	Home&Garden	consists of products related to home...
11	Novel	6	10	Books	encompasses written works, includin...
6	Jeans	3	100	Clothing	includes various types of apparel, in...
7	Shoes	4	60	Sports	features products associated with sp...
8	Bench	4	200	Sports	features products associated with sp...
9	Plants	5	40	Home&Garden	consists of products related to home...
1	Smartphone	1	1000	Electronics	includes electronic devices such as s...
4	Phone case	2	150	Accessories	encompasses additional items that c...
5	T-Shirt	3	40	Clothing	includes various types of apparel, in...
13	Skincare set	7	20	Beauty	includes cosmetics, skincare product...
14	Lipstick	7	5	Beauty	includes cosmetics, skincare product...

Figure 7 – View “Product_Category_View”

SQL code:

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "PRODUCT_CATEGORY_VIEW" ("PRODUCT_ID", "PRODUCT_NAME", "CATEGORY_ID", "PRICE", "CATEGORY_NAME", "CATEGORY_DESCRIPTION")
AS
SELECT p.product_id, p.product_name, p.category_id, p.price, c.category_name, c.category_description
FROM Products p
JOIN Categories c ON p.category_id = c.category_id;
```



REVIEW_CUSTOMER_VIEW

Columns | Code | Data | Errors | Grants | Dependencies | DDL

Columns... | Filter... | Count Rows | Download | Refresh

REVIEW_ID	RATING	COMMENTS	PRODUCT_NAME	CUSTOMER_ID	FIRST_NAME	LAST_NAME
2	4	Good job	Shoes	7	Michael	Davis
3	5	Thank you for your service!	Thermostat	4	Maria	Rodriguez
6	5	Well done!	Laptop	1	John	Doe
1	5	Excellent!	Headphones	2	Jane	Smith
4	5	Perfect!	Novel	5	Robert	White
5	4	Not bad, thanks	Jeans	6	Emily	Brown
7	5	Appreciate your work	Narrative	3	Alex	Johnson
8	3	Not totally satisfied	Headphones	4	Maria	Rodriguez
9	5	Simply perfect	Smartphone	6	Emily	Brown
10	5	I will recommend you!	Phone case	4	Maria	Rodriguez

Figure 8 – View “Review_Customer_View”

SQL code:

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "REVIEW_CUSTOMER_VIEW"
("REVIEW_ID", "RATING", "COMMENTS", "PRODUCT_NAME", "CUSTOMER_ID",
"FIRST_NAME", "LAST_NAME")
AS
SELECT r.review_id, r.rating, r.comments, p.product_name, c.customer_id, c.first_name,
c.last_name
FROM Reviews r
JOIN Products p ON r.product_id = p.product_id
JOIN Customers c ON r.customer_id = c.customer_id;
```

4. Implementation of Custom Database Functions

To extend the functionality of the application and enable reusable data operations, three custom database functions were developed:

- **Get_Product_Count_In_Category**: This function takes a category identifier as input and returns the total number of products within that category. It simplifies retrieving product counts for reporting or filtering purposes without requiring repeated complex queries.
- **Get_Review_Count**: This function accepts a product ID and returns the number of reviews associated with that product. It helps quickly assess customer engagement and feedback volume for individual products.
- **Get_Total_Order_Amount**: Designed to calculate the total monetary value of all orders placed by a specific customer, this function takes a customer ID as input and returns the sum of all order amounts. This is useful for generating customer purchase summaries and performing financial analysis.

Figure 9 – Function “Get_Product_Count_In_Category”

Figure 10 – “Function Get_Review_Count”

Figure 11 – Function “Get_Total_Order_Amount”

5. Development of Interactive Application Pages for Data Management

To provide a user-friendly interface for managing the database, five interactive pages were developed. Each page corresponds to one of the database tables and supports full CRUD (Create, Read, Update, Delete) operations, enabling users to

add new records, edit existing entries, and delete unwanted data. The pages include:

- **Categories Report**: Allows users to manage product categories, including adding new categories, modifying existing ones, and removing obsolete categories.
- **Customers Report**: Provides an interface for managing customer information, facilitating updates to customer details and the ability to maintain an up-to-date customer database.
- **Orders Report**: Enables users to view and manage order records, including creating new orders, editing order details, and deleting incorrect or cancelled orders.
- **Products Report**: Supports the management of product listings, allowing for the addition of new products, editing product information, and removing discontinued products.
- **Reviews Report**: Offers functionality to handle customer reviews, including adding feedback, editing review content, and deleting inappropriate or outdated reviews.

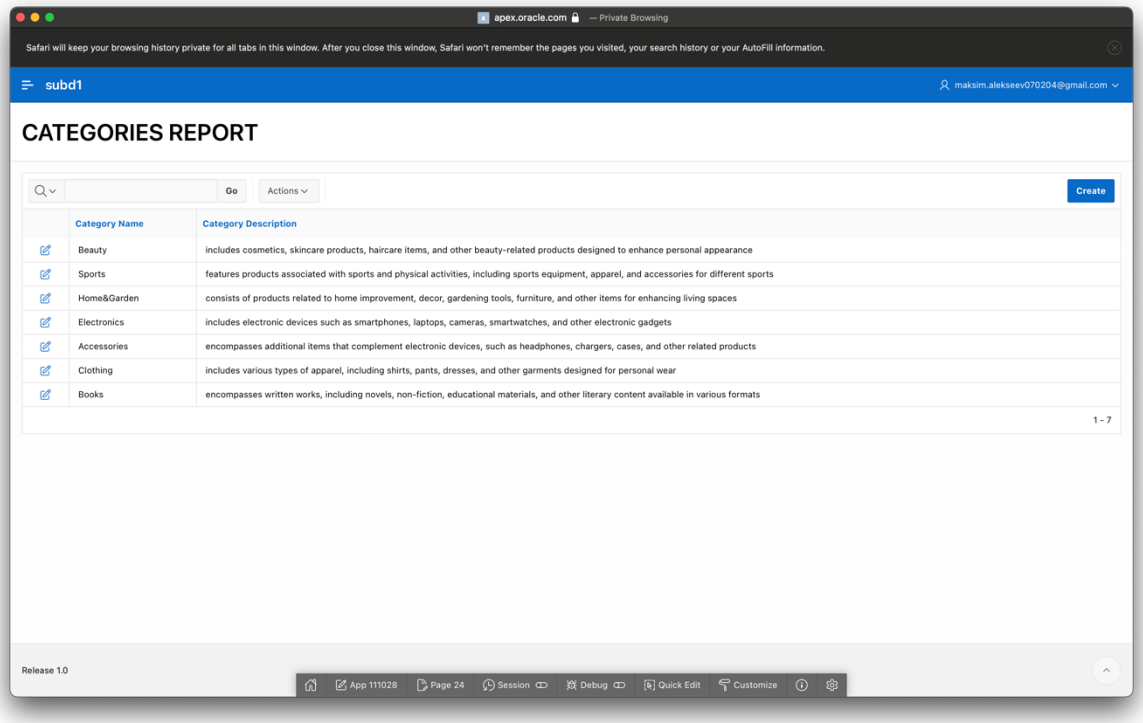


Figure 12 – The “Categories Report” page allows adding, editing, and deleting records from the “Categories” table.

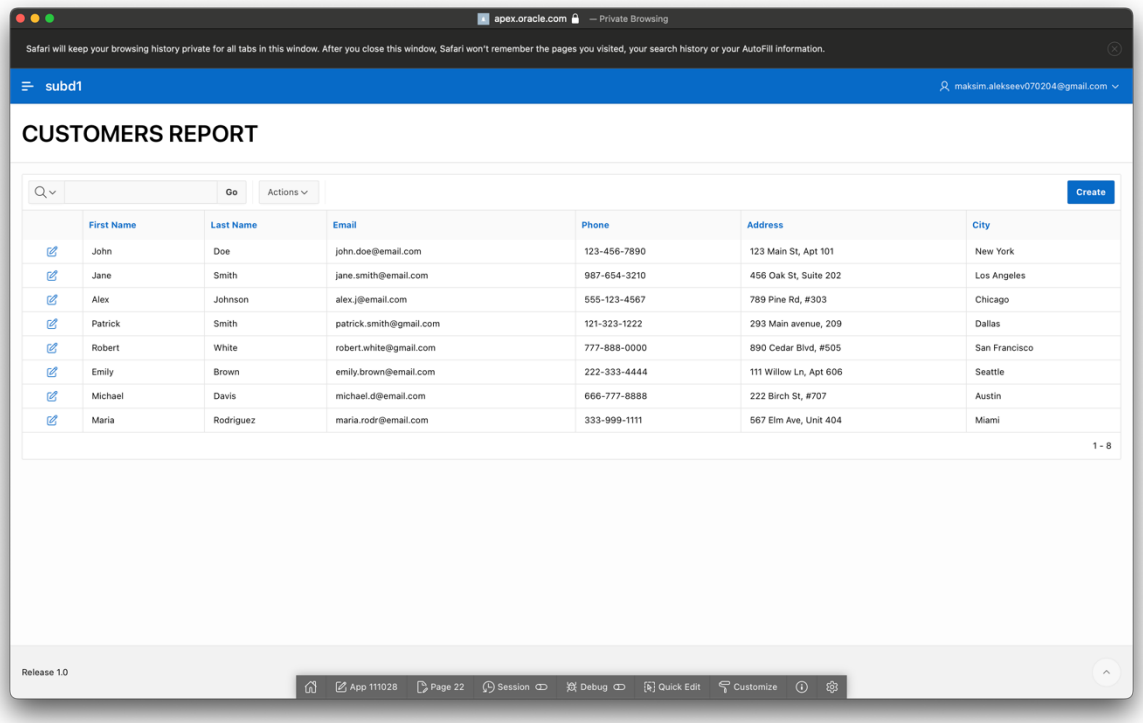


Figure 13 – The “Customers Report” page allows adding, editing, and deleting records from the “Customers” table.

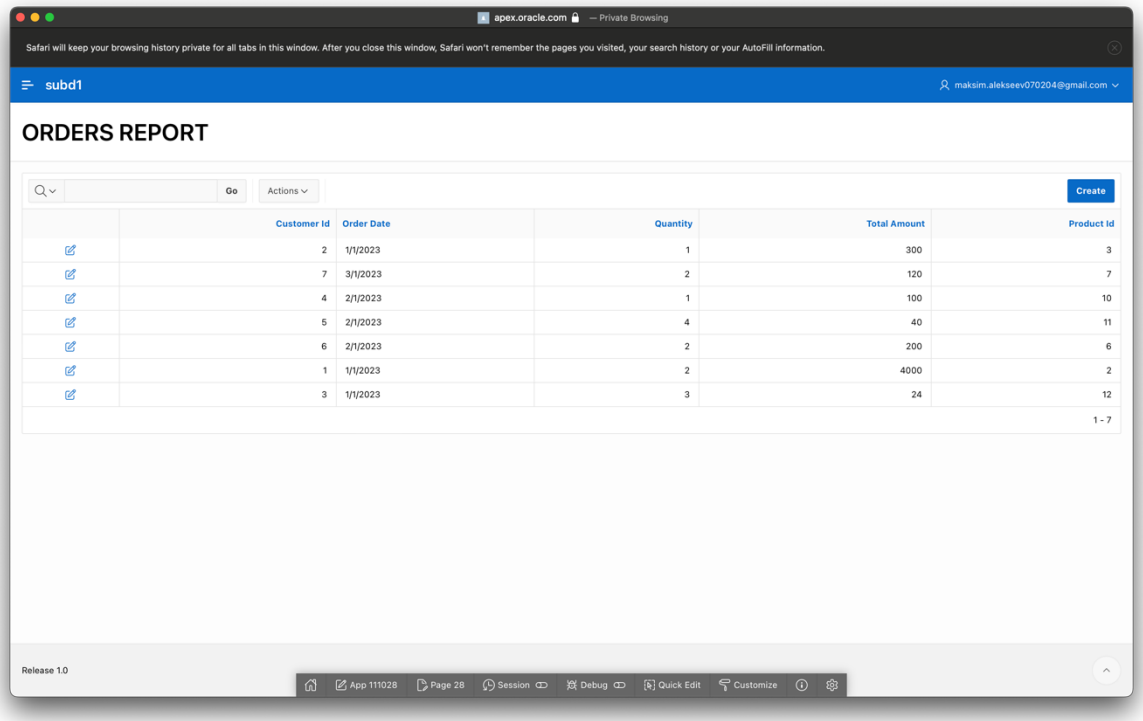


Figure 14 – The “Orders Report” page allows adding, editing, and deleting records from the “Orders” table.

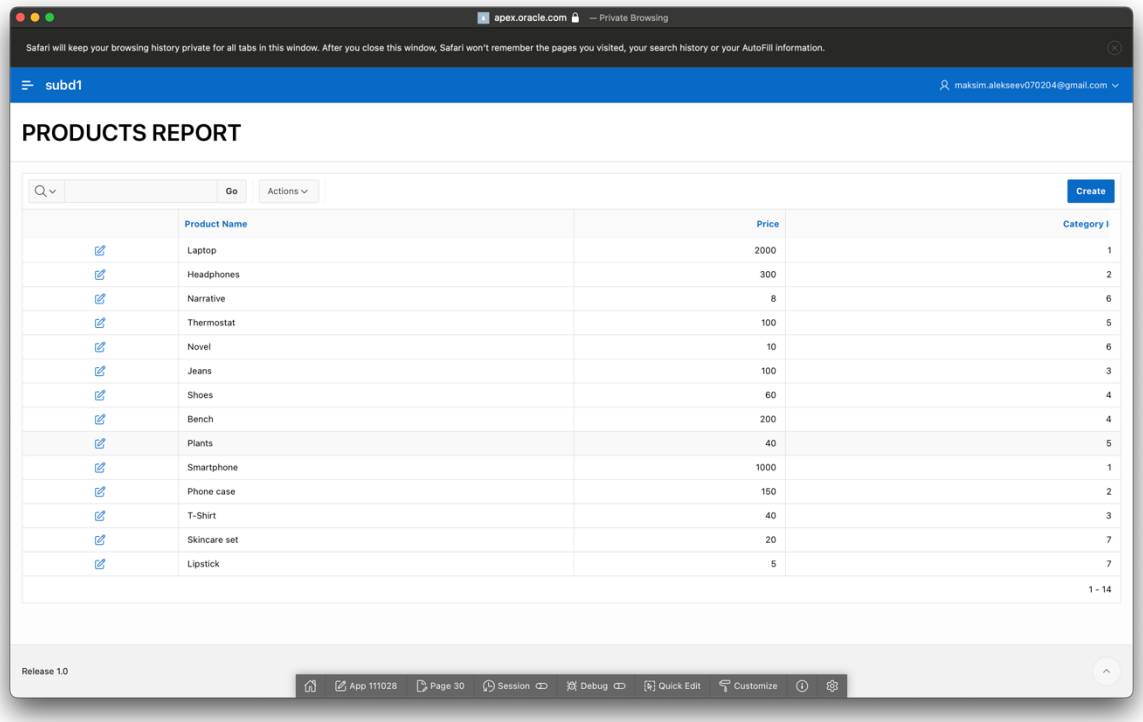


Figure 15 – The “Products Report” page allows adding, editing, and deleting records from the “Products” table

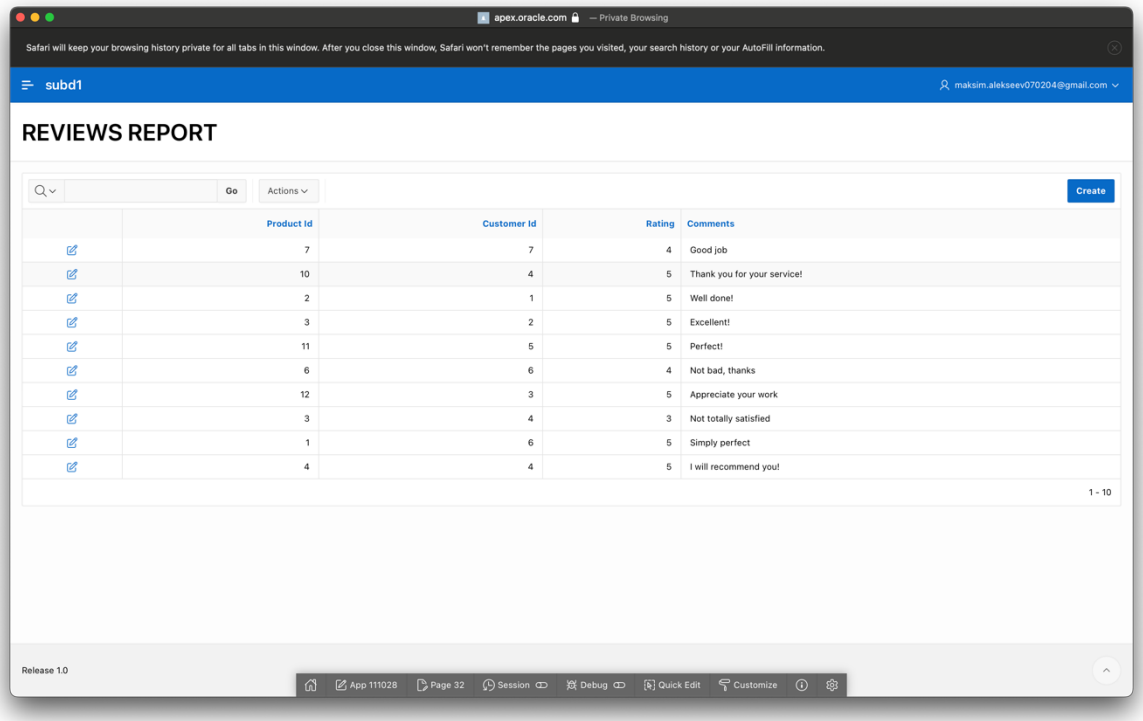


Figure 16 – The “Reviews Report” page allows adding, editing, and deleting records from the “Reviews” table.

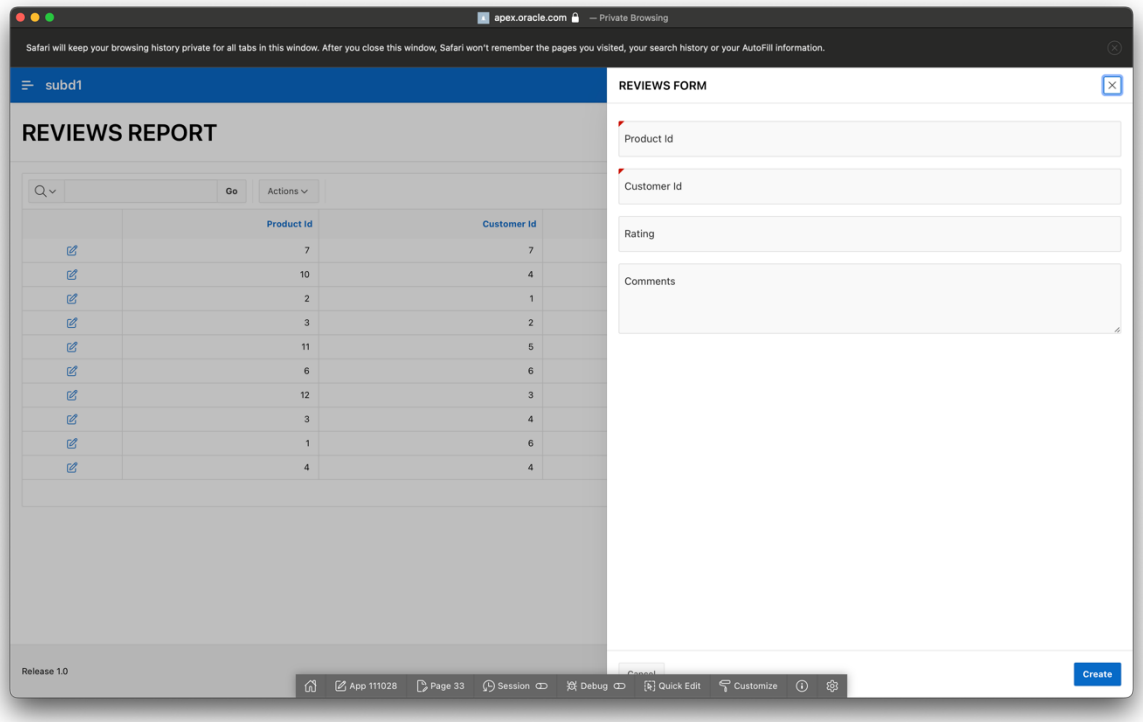


Figure 17 – Example of a form for adding records to the “Reviews” table.

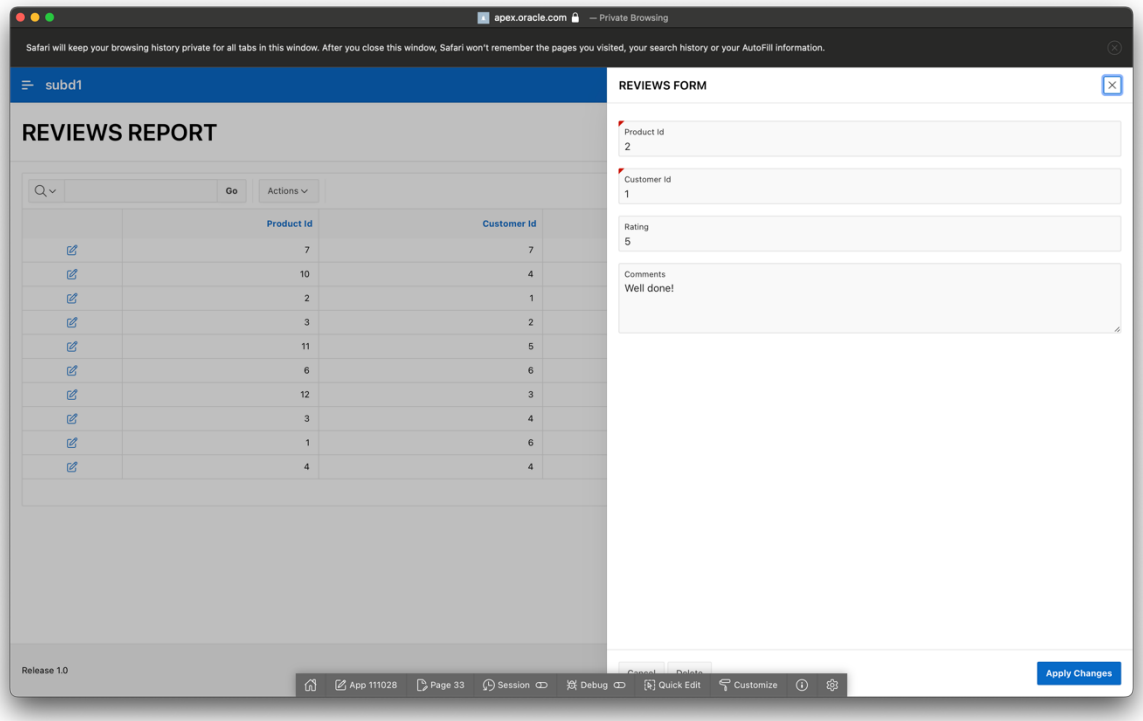


Figure 18 – Example of a form for editing and/or deleting records from the “Reviews” table.

6. Development of Data Display Pages with Filtering for Views

To enhance data visualization and user interaction, three dedicated pages were developed to display information retrieved from the previously created database views:

- Customer_Order_View.
- Product_Category_View.
- Review_Customer_View.

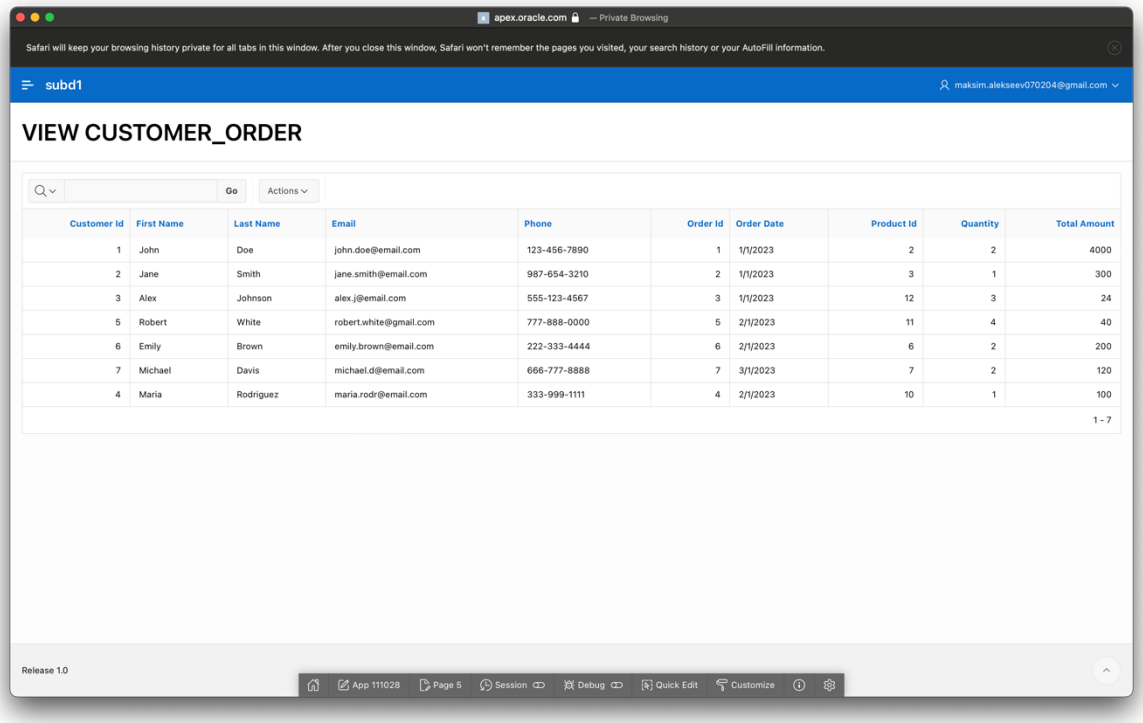


Figure 19 – The “Customer_Order View” page displays and filters data from the “Customer_Order_View” view.

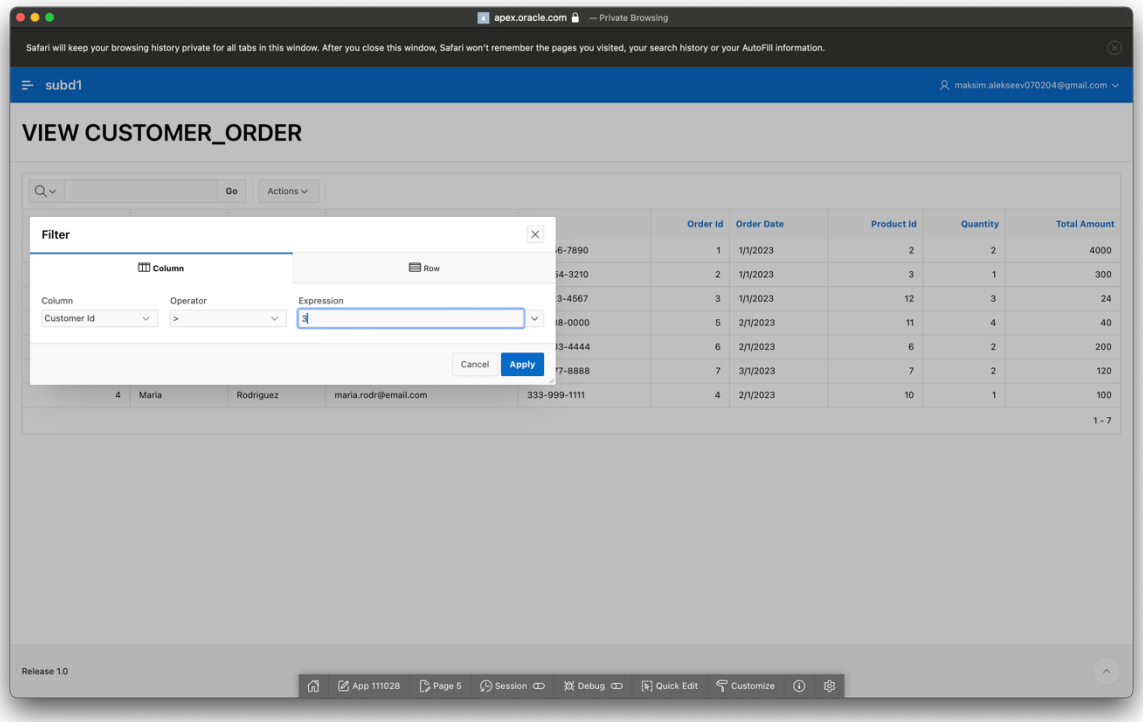


Figure 20 – Example of filtering data by the column value customer_id (customer_id > 3) from the “Customer_Order_View” view.

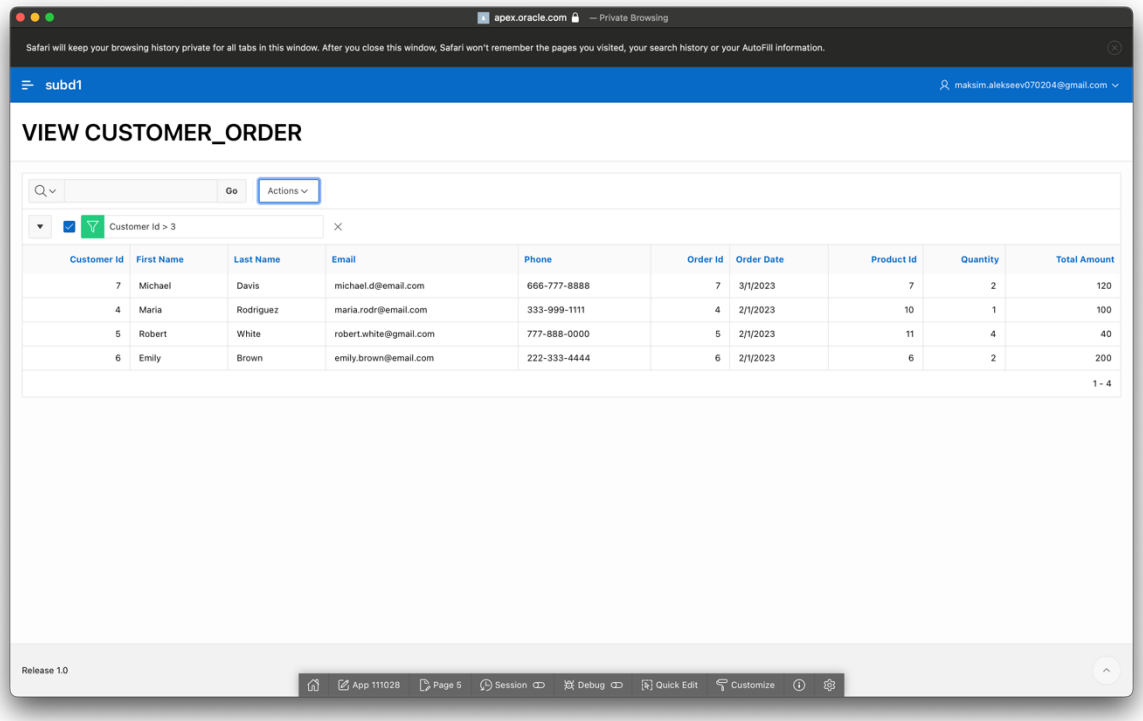


Figure 21 – The result of filtering data by the column value customer_id (customer_id > 3) from the “Customer_Order_View” view.

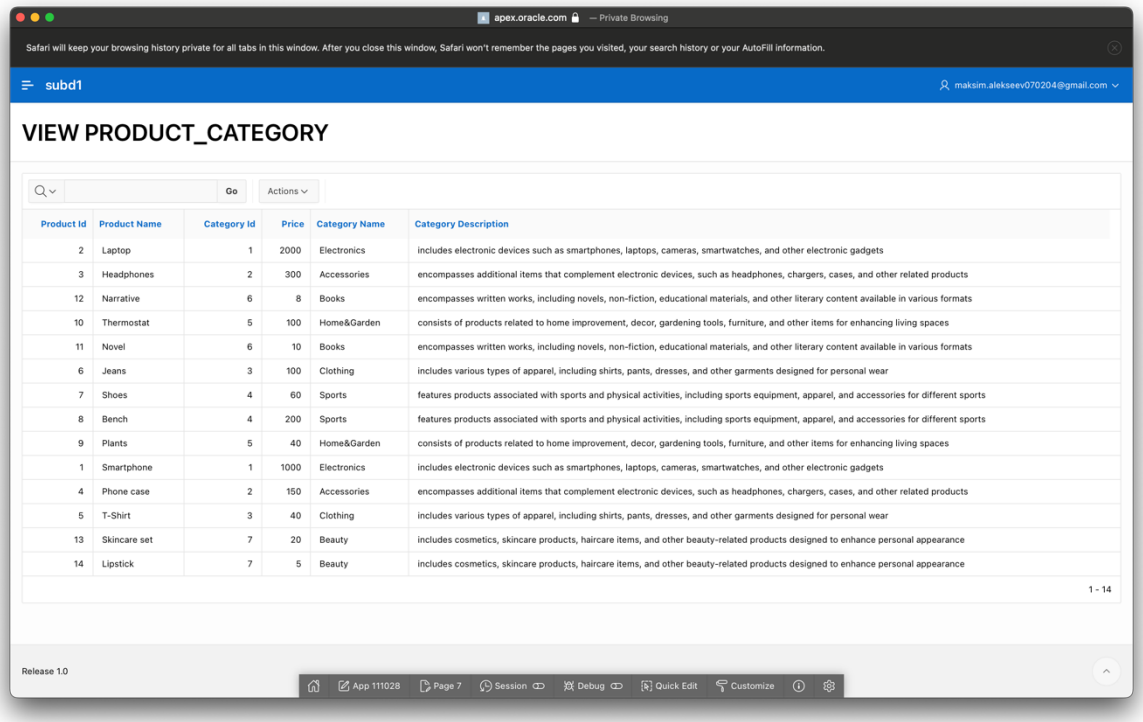


Figure 22 – The “Product_Category View” page displays and filters data from the “Product_Category_View“ view.

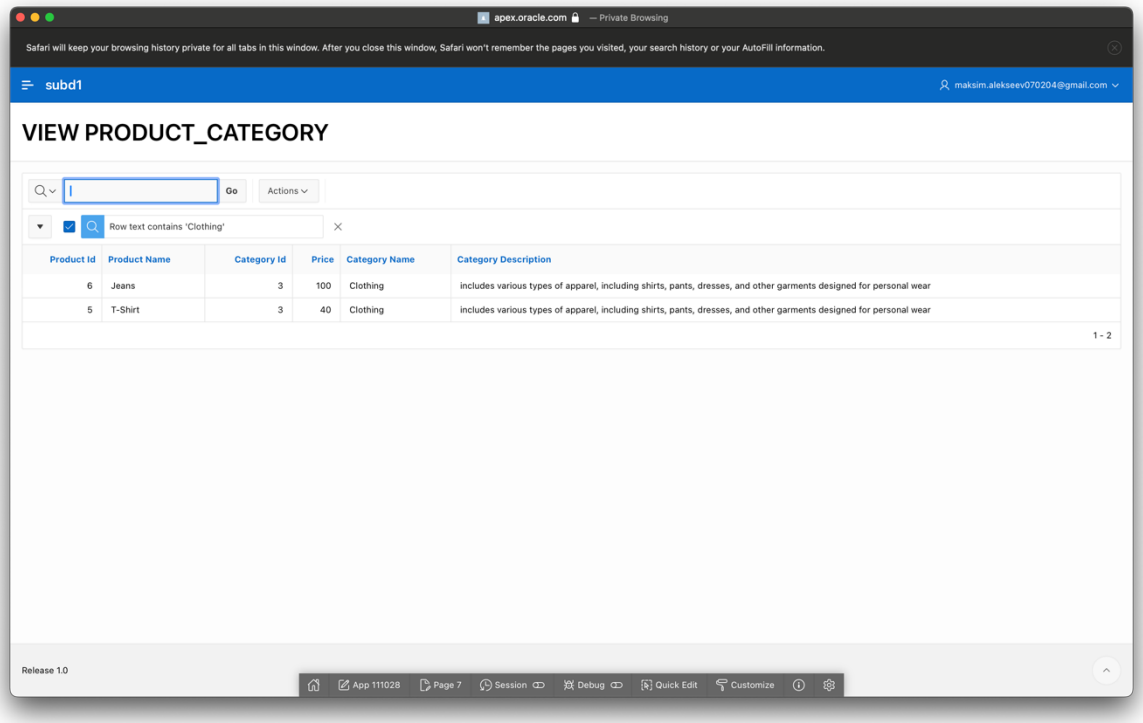


Figure 23 – Example of searching data by the keyword “Clothing” in the “Customer_Order_View” view.

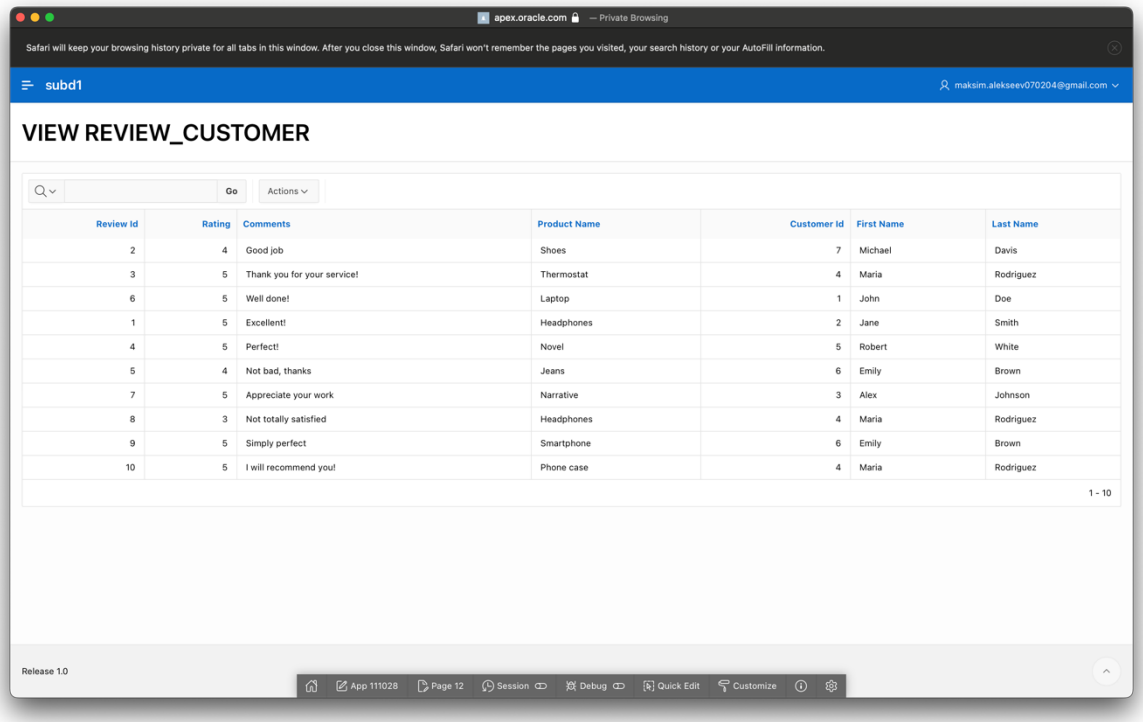


Figure 24 – The “Review_Customer View” page displays and filters data from the “Review_Customer_View” view.

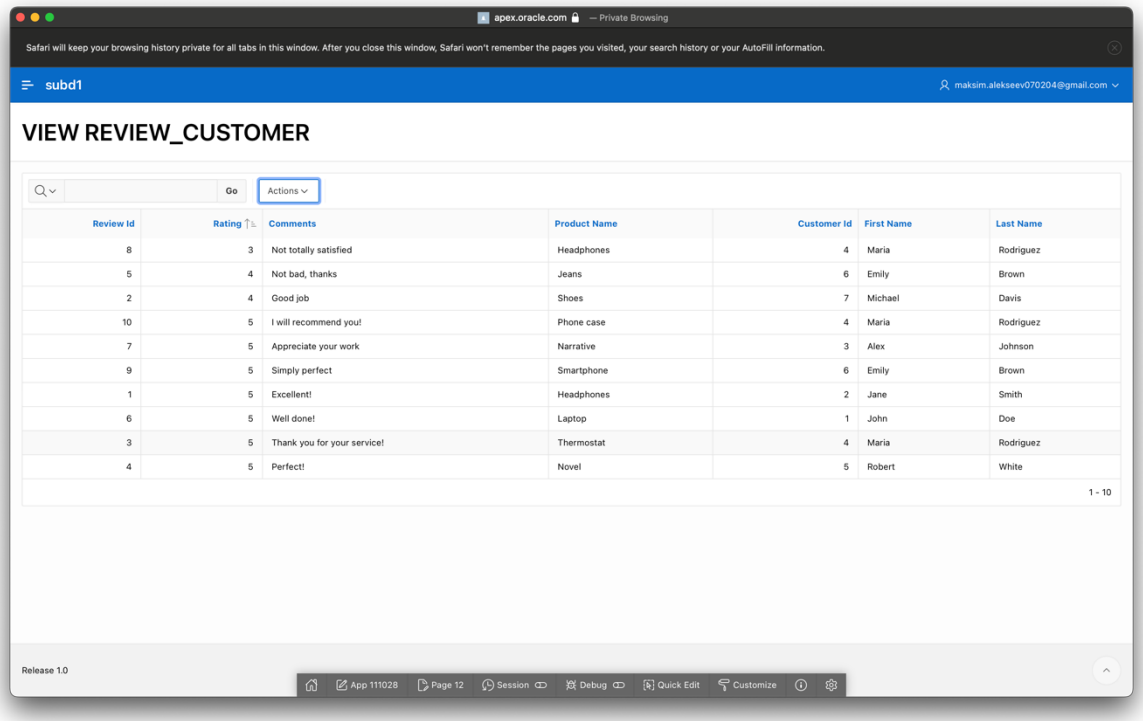


Figure 25 – Example of sorting data based on the “Rating” column in ascending order from the “Customer_Order_View” view.

7. Implementation of Authentication

Authentication was implemented on each of the previously created pages where data can be edited. Additionally, three separate pages without authentication were developed for external users:

- Products List – displays a list of all available products and their categories to the external user.
- Ratings Chart – shows a pie chart illustrating the distribution of ratings given by other users.
- Reviews Cards – displays “cards” with reviews from other users for the external user.

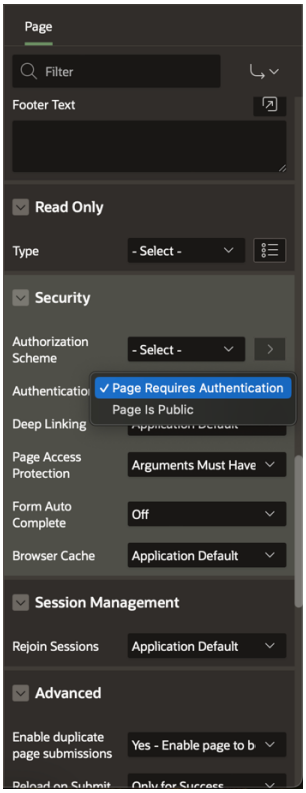


Figure 26 – Implementation of mandatory authentication on the “Categories Report” page.

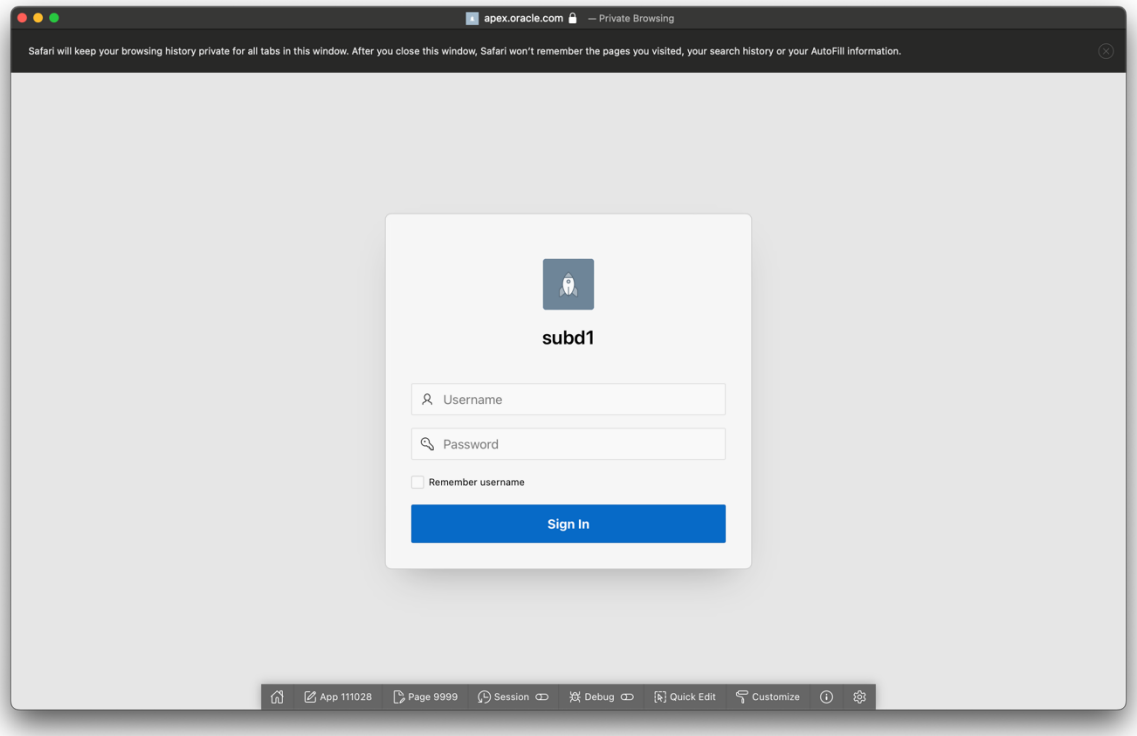


Figure 27 – Authentication prompt when accessing the “Categories Report” page.

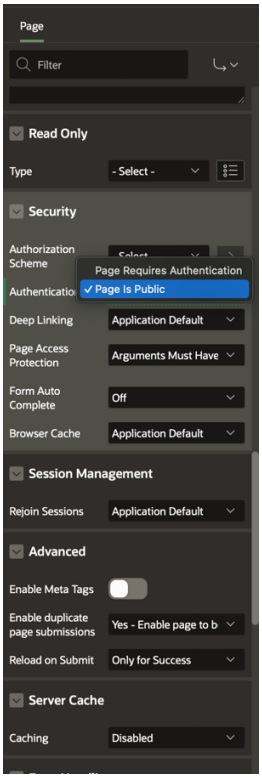


Figure 28 – Removal of mandatory authentication from the “Products List” page.

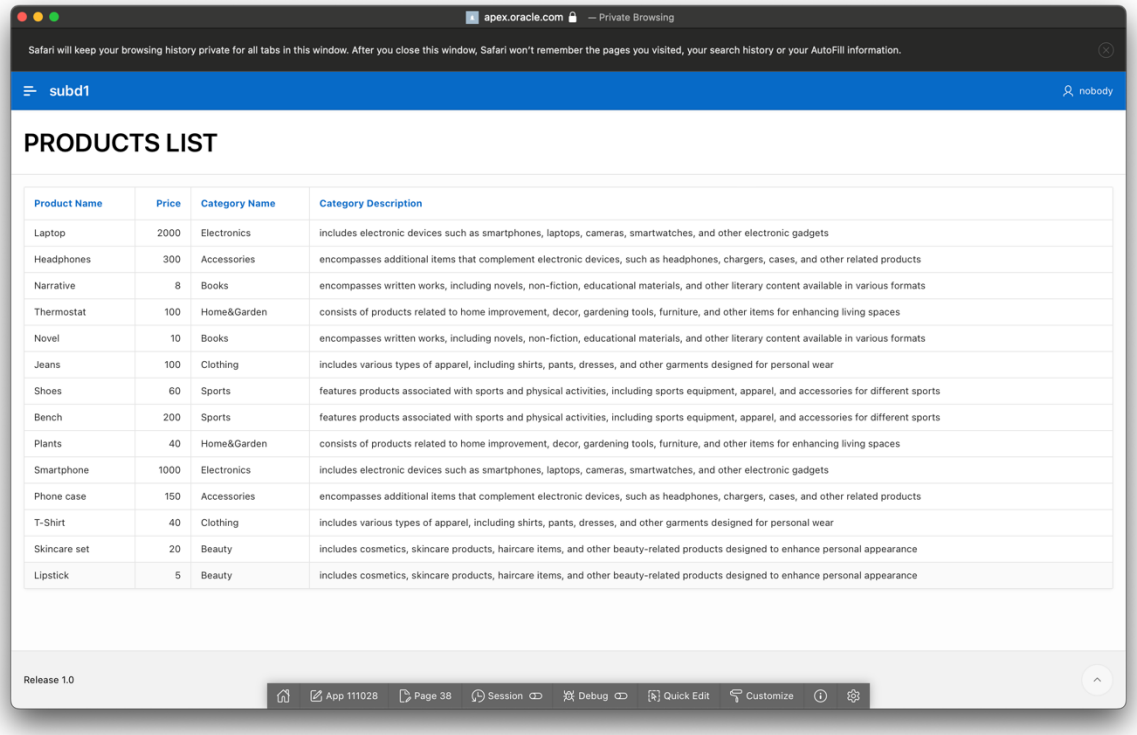


Figure 29 – The “Products List” page, accessible to external users.

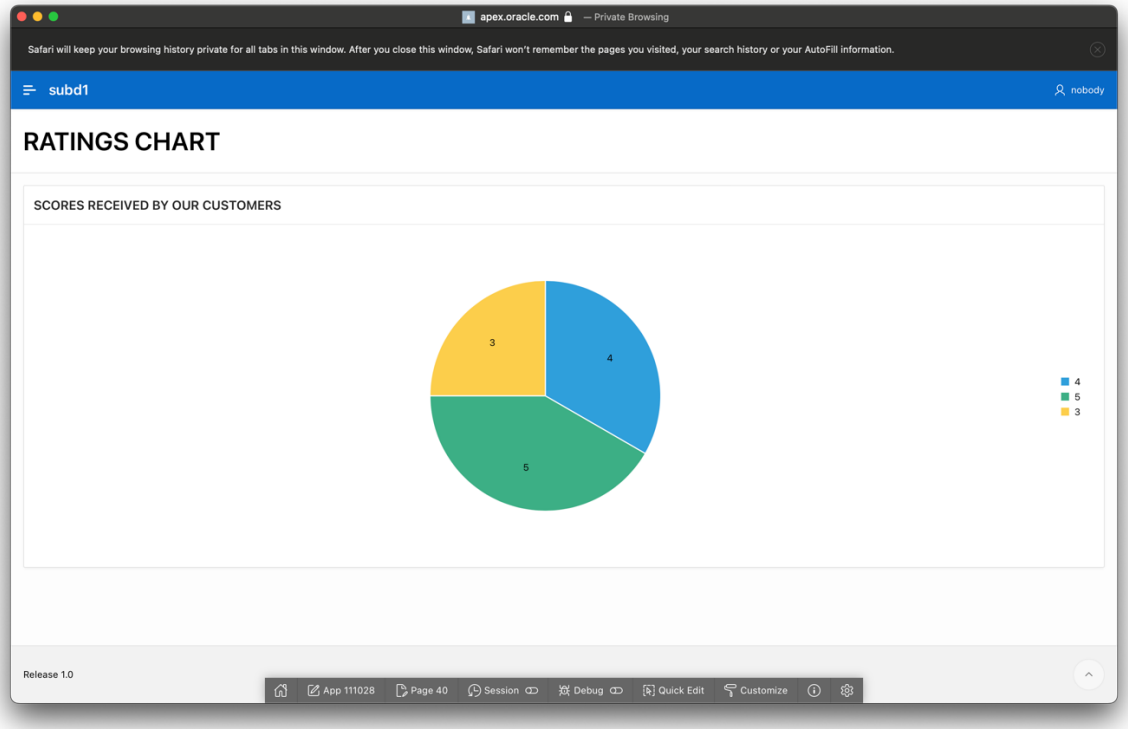


Figure 30 – The “Ratings Chart” page, accessible to external users.

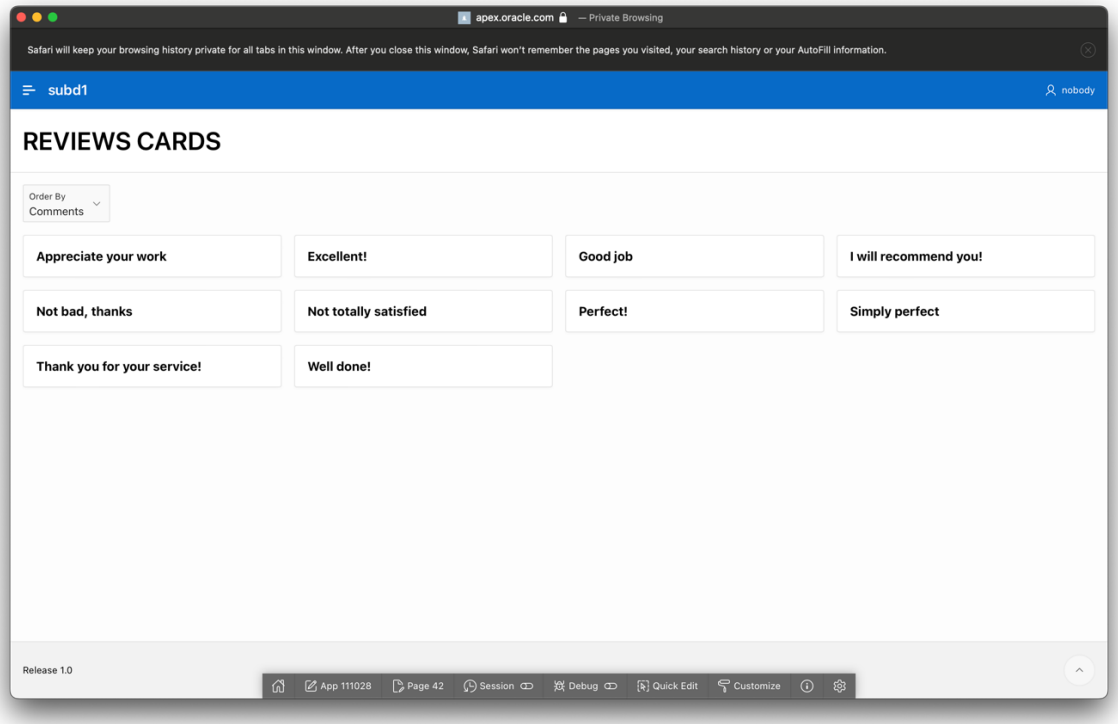


Figure 31 – The “Reviews Cards” page, accessible to external users.

Conclusions:

During the completion of this study, valuable hands-on experience was gained in developing applications using Oracle APEX technology:

- Five relational tables were created, linked with one-to-many relationships to ensure data integrity and logical connections.
- Three database views were developed to simplify complex queries and improve data accessibility.
- Three custom functions were implemented to encapsulate common calculations and data retrieval operations.
- Five interactive pages were built, enabling users to add, edit, and delete records in each database table.
- Three pages were designed to display data from the views, including filtering capabilities for efficient data exploration.
- Authentication-enabled pages were created to allow secure data modification by authorized users.
- Additional pages without authentication were provided for external users to access data in a read-only mode.