

ST. TERESA'S COLLEGE
(AUTONOMOUS)
AFFILIATED TO MAHATMA GANDHI UNIVERSITY



PROJECT REPORT ON
“ECO-SWAP”

In partial fulfillment of the requirements for the

Award of the degree of

**BVOC SOFTWARE
DEVELOPMENT**

By

ANN JOICE PETER

III BVOC Software Development

Register No: VB21SWD008

Under the guidance of

Asst. Prof HARSHA K.M

Department of Computer Applications

2021-2024

ST TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM
AFFILIATED TO MAHATMA GANDHI UNIVERSITY



CERTIFICATE

This is to certify that the practical record entitled "ECO-SWAP" is a bona fide record of the work done by ANN JOICE PETER (VB21SWD008) during the year 2021-2024 and submitted in partial fulfillment of the requirement for the degree of B. Voc Software Development under Mahatma Gandhi University.

Head of the Department



for
led
21/3/24
Internal Examiner

Penguin
22/3/2024
External Examiner

Date: 22-03-2024

DECLARATION

I, ANN JOICE PETER (Register no:VB21SWD008), B.VOC Software Development final year student at St. Teresa's College (Autonomous), Ernakulam, hereby declare that the project submitted named "ECO-SWAP"

for the Bachelors of Vocation Degree in Software Development is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

DATE: 22nd March 2024

PLACE: ERNAKULAM



ANN JOICE PETER

ACKNOWLEDGEMENT

First and foremost, I would like to thank God almighty for the successful completion of my project. I express my sincere thanks to **Provincial Superior and Manager Rev. Dr. Sr. Vinitha CSST, Rev. Sr. Emeline CSST and Principal Dr. Alphonsa Vijaya Joseph of St. Teresa's college (AUTONOMOUS)** for giving me an opportunity to undertake this project. I express my sincere gratitude to the **Head of the department, Ms. Remya C.J** and extend my sincere gratitude to **Asst. Prof. Harsha K.M**, my project guide for her constant support which helped in the successful completion of my project. I'm grateful to all the faculties of the Department of Computer Applications for their valuable help and guidance during each stage of my project. I would like to thank my parents and friends for motivating me and providing me with the right environment to make this project a great success.

ANN JOICE PETER

SYNOPSIS

An **ECO SWAP** is a service in which the seller can sell eco-friendly products and buyers can buy products via the internet. The system is secured because only authorized users can enter the system using the username and password. It enables users to access Eco Friendly products online, anytime and anywhere, through electronic devices like computers, tablets, and smartphones. Developed using PHP, MySQL, JavaScript, HTML, and CSS, our Eco-Swap web application is user- friendly, with separate roles for administrators, users and buyers. Shops are managed and added by admin after verification,

CONTENTS

| | | |
|-------|---|----|
| 1 | INTRODUCTION..... | 1 |
| 1.1 | About the project..... | 2 |
| 2 | SYSTEM ANALYSIS | 3 |
| 2.1 | Introduction | 4 |
| 2.2 | Existing System..... | 4 |
| 2.3 | Proposed System | 5 |
| 2.4 | System Specification | 6 |
| 2.5 | Operating System | 7 |
| 2.6 | Language or Software Package | 7 |
| 2.7 | Hardware and Software Specification | 8 |
| 3 | SYSTEM DESIGN | 9 |
| 3.1 | Introduction..... | 10 |
| 3.2 | Dataflow Diagram..... | 10 |
| 3.3 | Database Design..... | 14 |
| 4 | SYSTEM DEVELOPMENT | 18 |
| 4.1 | Introduction..... | 19 |
| 4.2 | Process description... .. | 19 |
| 5 | SYSTEM TESTING AND IMPLEMENTATION..... | 20 |
| 5.1 | Introduction..... | 21 |
| 5.2 | Debugging..... | 22 |
| 5.2.1 | Black box testing..... | 22 |
| 5.2.2 | White box Testing | 22 |
| 5.2.3 | System Security | 22 |
| 5.2.4 | Scope for Future Enhancement | 22 |
| 6 | CONCLUSION..... | 23 |
| 7 | APPENDIX..... | 25 |
| 7.1 | Input & Output Screen..... | 26 |
| 8 | BIBLIOGRAPHY | 38 |

1. INTRODUCTION

INTRODUCTION

Eco- swap project aims to be part of sustainable development that helps users to create a change on products they use, without destroying nature. An Eco-Swap is a service in which the seller can sell eco-friendly. Products and buyers can buy products via the internet. The system is secured because only authorized users can enter the system using the username and password. It enables users to access Eco Friendly products online, anytime and anywhere, through electronic devices like computers, tablets, and smartphones. Developed using PHP, MySQL, JavaScript, HTML, and CSS, our Eco-Swap

1.1 ABOUT PROJECT

Eco-Swap provides various collections of eco-friendly products and different shops which have a variety of products which can be used repeatedly. Shop owners can add their new products and learning technologies allow access to the latest and best information available. E-learning information means our education sources are always relevant and updated. Technology can provide students with access to large amounts of online resources and information. This allows and encourages them to carry out their own research and therefore become more independent. Technology can also simplify learning by making complex concepts more digestible for different learning styles. Technology is helpful in holding attention which allows students to better absorb information. In e-learning, the admin module focuses on managing course content, user accounts. The user module focuses on providing an intuitive interface for the best shopping, quizzes, and educational content.

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

System Analysis is a detailed study of the various operations performed by the system and their relationship within the modules of the system. This phase involves the study of the parent system and identification of the system objectives. The main objective of this phase involves gathering necessary information and using the structured tool for analysis. This includes designing the system. In this project, the requirements are studied in detail and information are collected and documented.

2.2 EXISTING SYSTEM

There are few online stores selling eco -friendly products but they are not specifically for eco-friendly products . some of them are Flipkart, amazon, Myntra

2.3 PROPOSED SYSTEM

The proposed system is an innovative eco-friendly web application designed to create a sustainable marketplace connecting sellers of eco-friendly products with environmentally conscious buyers. By providing a dedicated platform for sellers to list and sell a variety of eco-friendly items, and for buyers to discover and purchase these products, the system aims to foster a community committed to sustainable living and consumption.

Objectives of the Eco-Swap project is:

- promote sustainable
- Educational
- Convenience and Flexibility
- Reduce usage of plastic

2.4 SYSTEM SPECIFICATION

System specification specifies the hardware and software configuration of the new system. It helps to define the operational and performance guidelines for a system. Eco-Swap is a web application developed in WAMP, SQLYOG, VS CODE, ANDROID STUDIO, and PYTHON DJANGO for selling, buying eco-friendly product and helps in sustainable development

2.5 OPERATING SYSTEM

Windows is a series of operating systems developed by Microsoft.

Each version of Windows includes a graphical user interface, with a desktop that allows users to view files and folders in windows.

Microsoft introduced an operating environment named Windows on November 20, 1985 as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs). For the past two decades, windows has been the most widely used operating system for personal computer PCs.

2.6 LANGUAGE OR SOFTWARE PACKAGE

The application is built using Django, which is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development.

SQL Yog is a popular graphical user interface (GUI) tool for managing MySQL and MariaDB databases. It provides database administrators and developers with a convenient and intuitive interface for performing various database management tasks such as creating, editing, and deleting databases, tables, and queries.

Bootstrap is a free front-end framework for faster and easier web development.

Bootstrap includes HTML and CSS based design templates for

7

typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins Bootstrap also gives you the ability to easily create responsive designs.

2.7 HARDWARE & SOFTWARE SPECIFICATIONS

- ❖ Front End

- ❖ HTML5, CSS and Bootstrap

- ❖ Back End:

Database Base Management System: Mysql,python Django

- ❖ Operating System:

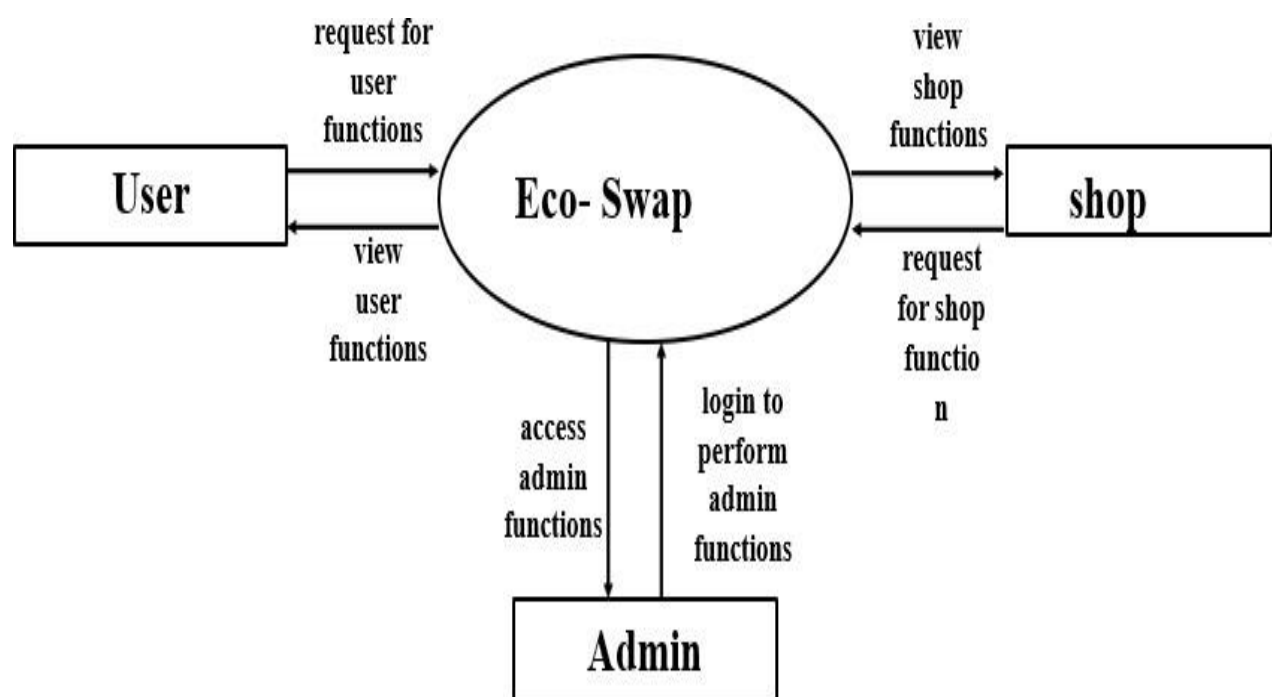
- Microsoft Windows 7 or later
- Browser: Google Chrome

3. SYSTEM DESIGN

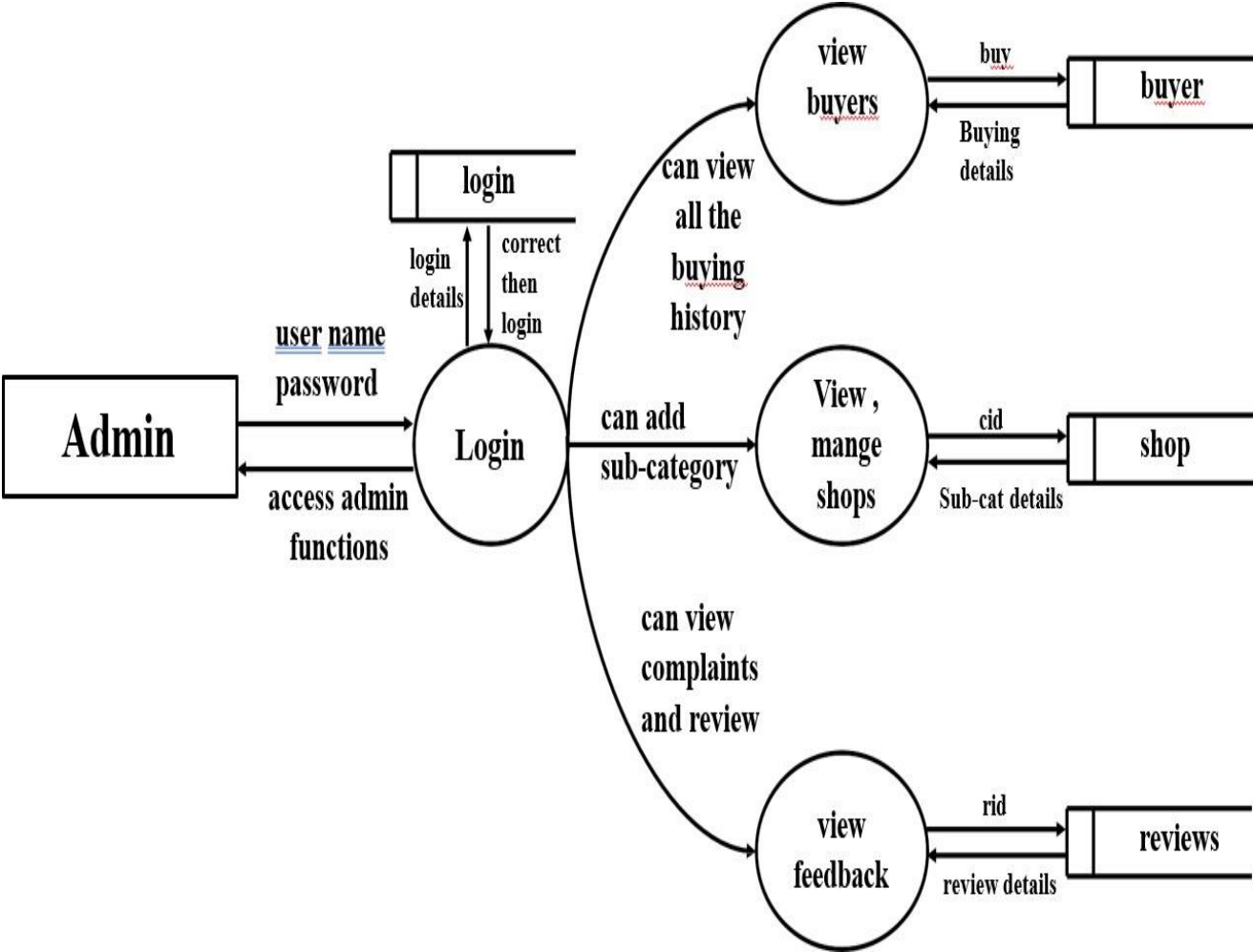
3.1 INTRODUCTION

A system architecture or system's architecture is the conceptual model that defines the structure, behavior, and more views of a system. System architecture can comprise system components, the externally visible properties of those components, and the relationships between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs). The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people. System architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user. The structural design reduces complexity, facilitates change and result in easier implementation by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirement. The system is based on the three-tier architecture. The first level is the user interface (presentation logic), which displays controls, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or database. It contains logic about to retrieve and update data. The important feature about the three-tier design is that information only travels from one level to an adjacent level UTO.

3.2 DATA FLOW DIAGRAM

LEVEL 0:

LEVEL 1:Admin



3.2 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

- Ease of learning and use.
- Controlled redundancy.
- Data independence.
- More information at low cost.
- Accuracy and integrity.
- Recovery from failure.
- Privacy and security.
- Performance.

TABLE DESIGN

Table 1: Admin table

| Fieldname | Type | Size | Description |
|-----------------|----------|-------|-------------|
| id | integer | (11) | Primary Key |
| action time | datetime | (6) | |
| object_id | longtext | | foreign key |
| object_repr | varchar | (200) | |
| Action_flag | smallint | (5) | |
| Change_message | longtext | | |
| Content_type_id | integer | (11) | |
| User_id | integer | (11) | foreign key |

Table 2: e_app_buyer:

| Fieldname | Type | Size | Description |
|-----------|---------|------|-------------|
| id | bigint | 20 | NOT NULL |
| firstname | varchar | 100 | NOT NULL |
| lastname | varchar | 100 | NOT NULL |
| district | varchar | 100 | NOT NULL |
| place | varchar | 100 | NOT NULL |
| post | varchar | 100 | NOT NULL |
| pincode | varchar | 100 | NOT NULL |
| email | varchar | 100 | NOT NULL |
| phone | varchar | 10 | NOT NULL |
| Login_id | bigint | 20 | NOT NULL |

Table 3:e_app_payment:

| id | amount | Buyer_id | Ordermain_id | date |
|----|--------|----------|--------------|------------|
| 1 | 1000 | 4 | 1 | 2024-02-07 |
| 2 | 2400 | 4 | 2 | 2024-02-13 |
| 3 | 240 | 21 | 3 | 2024-02-15 |
| 4 | 120 | 24 | 4 | 2024-02-15 |
| 5 | 820 | 27 | 5 | 2024-02-12 |
| 6 | 400 | 27 | 6 | 2024-02-12 |

4. SYSTEM DEVELOPMENT

4.1 INTRODUCTION

Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software.

4.2 PROCESS DESCRIPTION

- ❑ In the project, there are three types of users: -
 - Administrator (Admin)
 - Seller
 - Buyer

1. Administrator:

- Responsible for seller and buyer management
- Manages the whole System.
- Updating details and system.
- Respond to all the complaint.
- Approves buyer's request for creating account on application.
- Can Create Update Read and Delete

2. Seller:

- Sign up.
- Choose selling products.

- Can upload pictures for the item they want to sell.
- Can do selling

3. Buyer:

- Sign up.
- Choose buying product
- Can click on the item and add to cart.
- Can see the pictures uploaded by the sellers.

.

5. SYSTEM TESTING & IMPLEMENTATION

5.1 INTRODUCTION

This chapter deals with implementation of a software as well as its maintenance. Implementation allows the users to take over its operation for use and evaluation. Maintenance changes the existing system, enhancement adds features to the existing system, and development replaces the existing system.

DEBUGGING (BLACK & WHITE BOX TESTING)

5.1.1 BLACK BOX TESTING:

Black-box testing is a type of software testing in which the tester is not connected with the internal knowledge or implementation details of the software, but rather validating the functionality based on the provided specifications or requirements.

5.1.2 WHITE BOX TESTING:

White box testing techniques analyse the internal structures the used data structures, internal design, code structure, and the working of the software rather than just the functionality as in black-box testing. It is also called glass box testing or clear box testing or structural testing. It is used to test the software's internal logic, flow, and structure. The tester creates test cases to examine the code paths and logic flows to ensure they meet the specified requirement.

5.1.3 SYSTEM SECURITY:

Security testing is essential for software that processes confidential data to prevent system intrusion by hackers

5.1.4 SCOPE FOR FUTURE ENHANCEMENT:

In the realm of future enhancements, Eco Swap holds immense potential for further evolution and refinement. Community engagement features could be introduced, such as forums or discussion boards, nurturing a sense of camaraderie among users united by their commitment to sustainability. Personalization could be deepened, offering tailored product recommendations based on individual preferences and purchase history, thereby enhancing user satisfaction and loyalty. The development of a dedicated mobile application would extend the platform's accessibility, empowering users to engage with eco-friendly commerce seamlessly across devices. Moreover, global expansion presents a promising avenue for Eco Swap, with opportunities to collaborate with international eco-friendly businesses and organizations, thus broadening its impact on a global scale. By continuously innovating in areas such as supply chain transparency, augmented reality integration, and data analytics, Eco Swap can remain at the forefront of sustainable commerce, inspiring positive environmental change and fostering a community dedicated to eco-conscious living.

6. CONCLUSION

CONCLUSION

In conclusion, Eco Swap represents a pioneering effort in fostering sustainable commerce by connecting sellers and buyers of eco-friendly products. With its user-friendly interface, robust features, and commitment to environmental consciousness, Eco Swap not only facilitates transactions but also builds a community dedicated to sustainable living. As we move forward, Eco Swap remains poised to drive positive change and inspire individuals to make eco-friendly choices, ultimately contributing to a more sustainable future for all.

.

7. APPENDIX

7.1 INPUT AND OUTPUT SCREEN:

HOME PAGE:

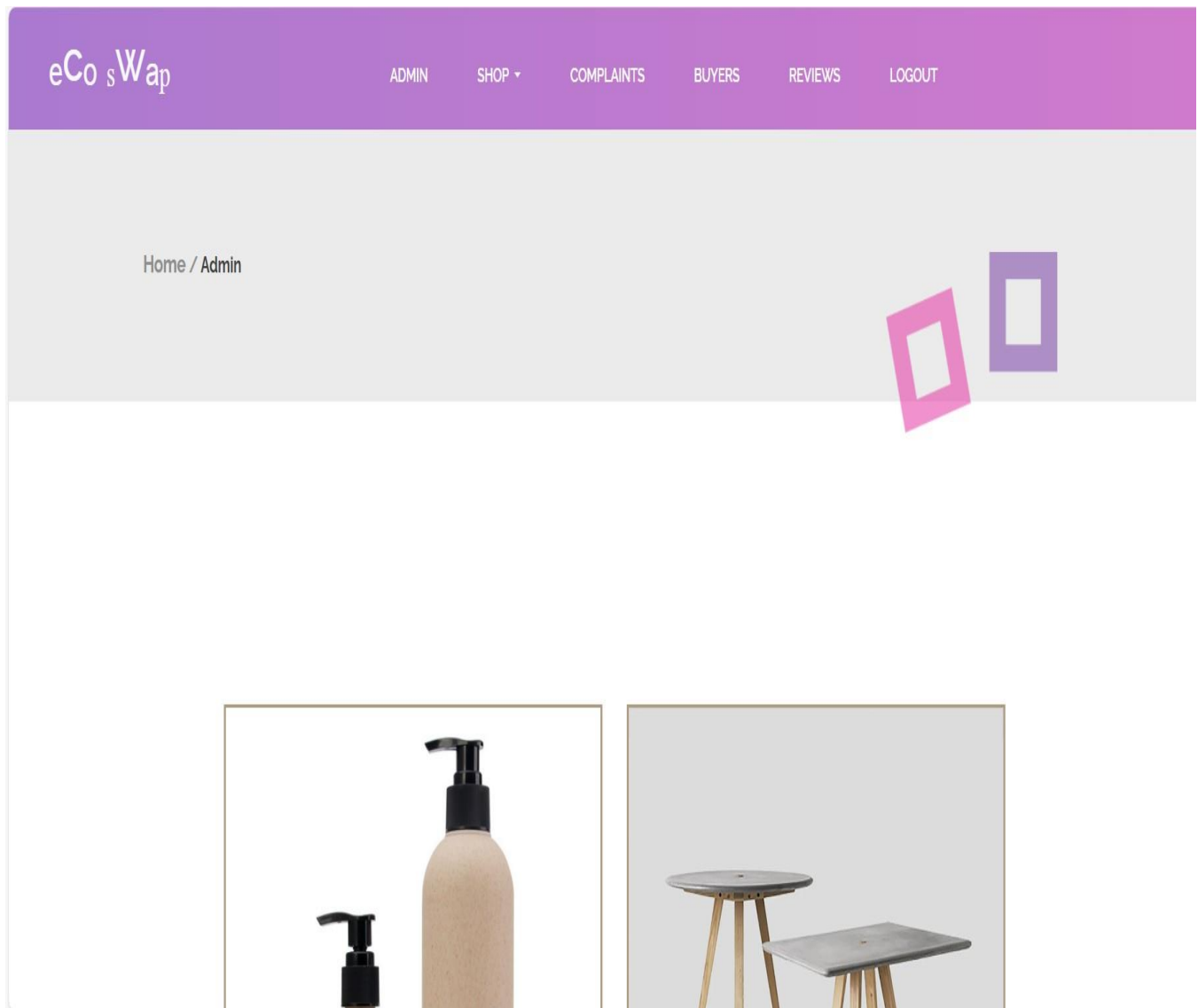


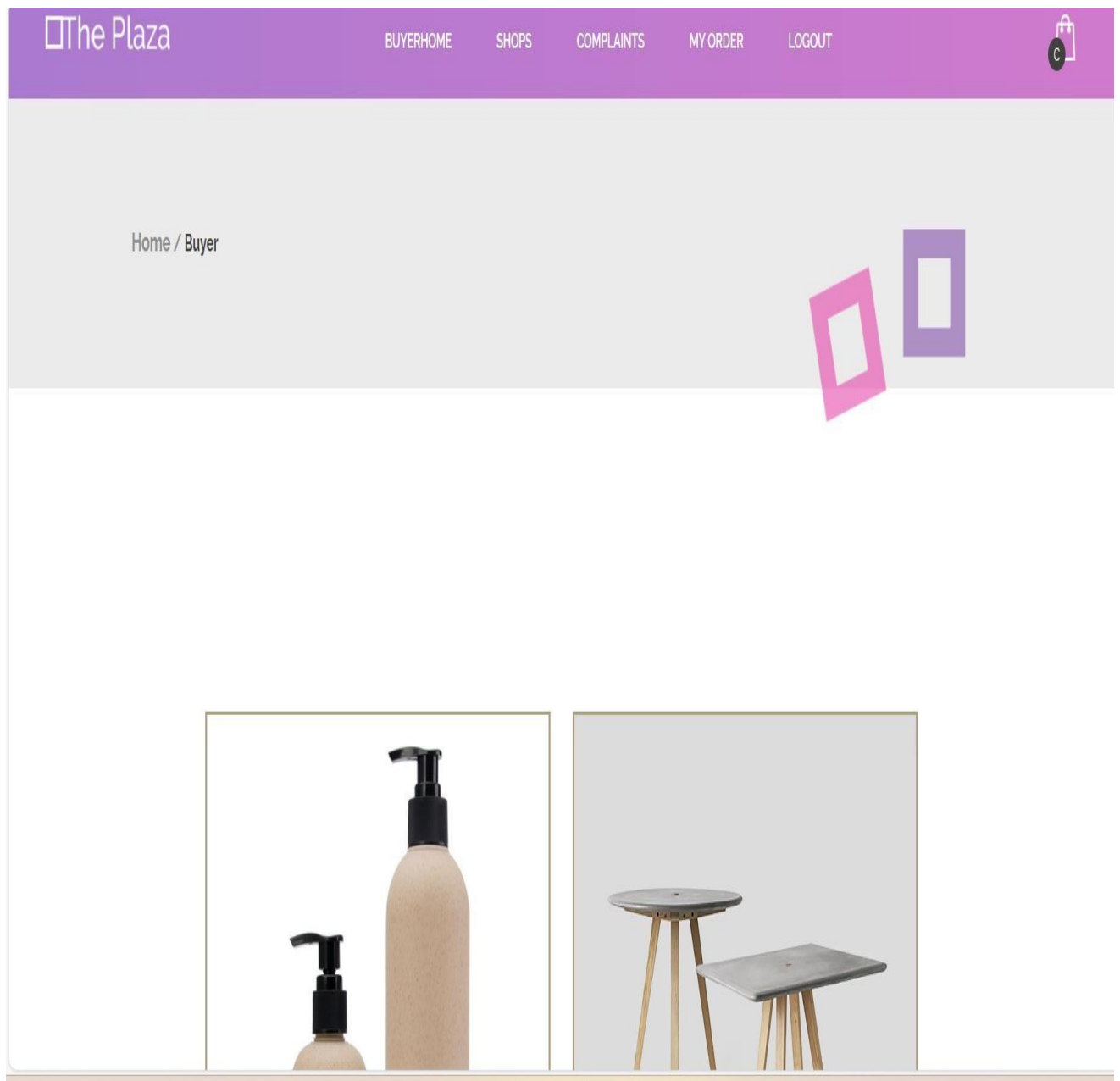
VENDOR LOGIN PAGE:

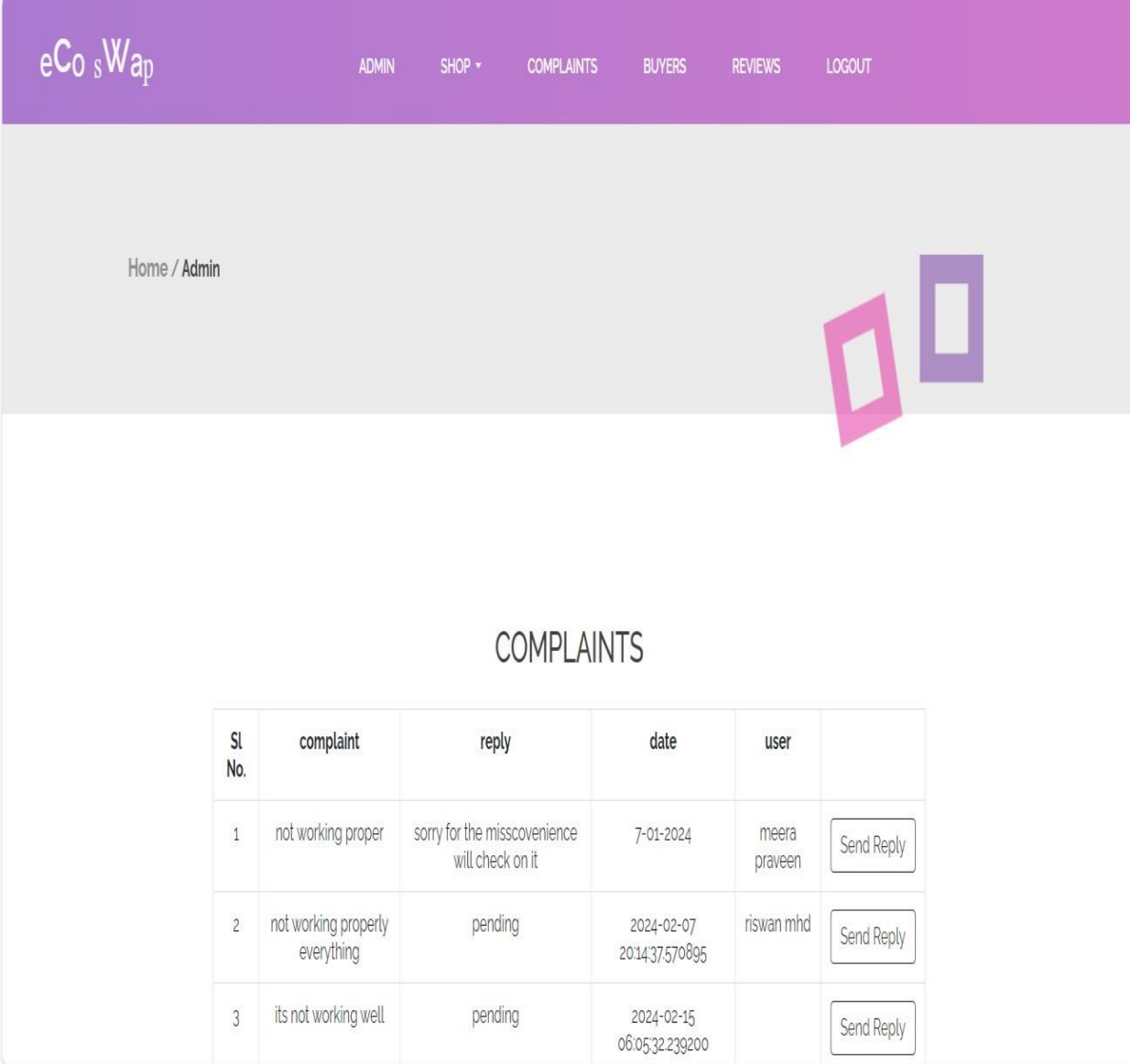
VENDOR SIGNUP

| | |
|---------------------------------------|--------------------------|
| shopname | <input type="text"/> |
| place | <input type="text"/> |
| landmark | <input type="text"/> |
| phone | <input type="text"/> |
| email | <input type="text"/> |
| Username | <input type="text"/> |
| Password | <input type="password"/> |
| <input type="button" value="SIGNUP"/> | |

BUYER LOGIN PAGE:

ADMIN HOME:

USER HOME:

USER COMPLAINTS:


Home / Admin

COMPLAINTS

| St No. | complaint | reply | date | user | |
|--------|------------------------------------|--|-------------------------------|------------------|----------------------------|
| 1 | not working proper | sorry for the misscovenience will check on it | 7-01-2024 | meera praveen | Send Reply |
| 2 | not working properly everything | pending | 2024-02-07 20:14:37.570895 | riswan mhd | Send Reply |
| 3 | its not working well | pending | 2024-02-15 06:05:32.239200 | | Send Reply |

USER PAYMENT SETTINGS:

Account Number

546

Card Number

345

CVV Number

354

Pin Number

342

Account Holder

3w4

Amount:

820.0

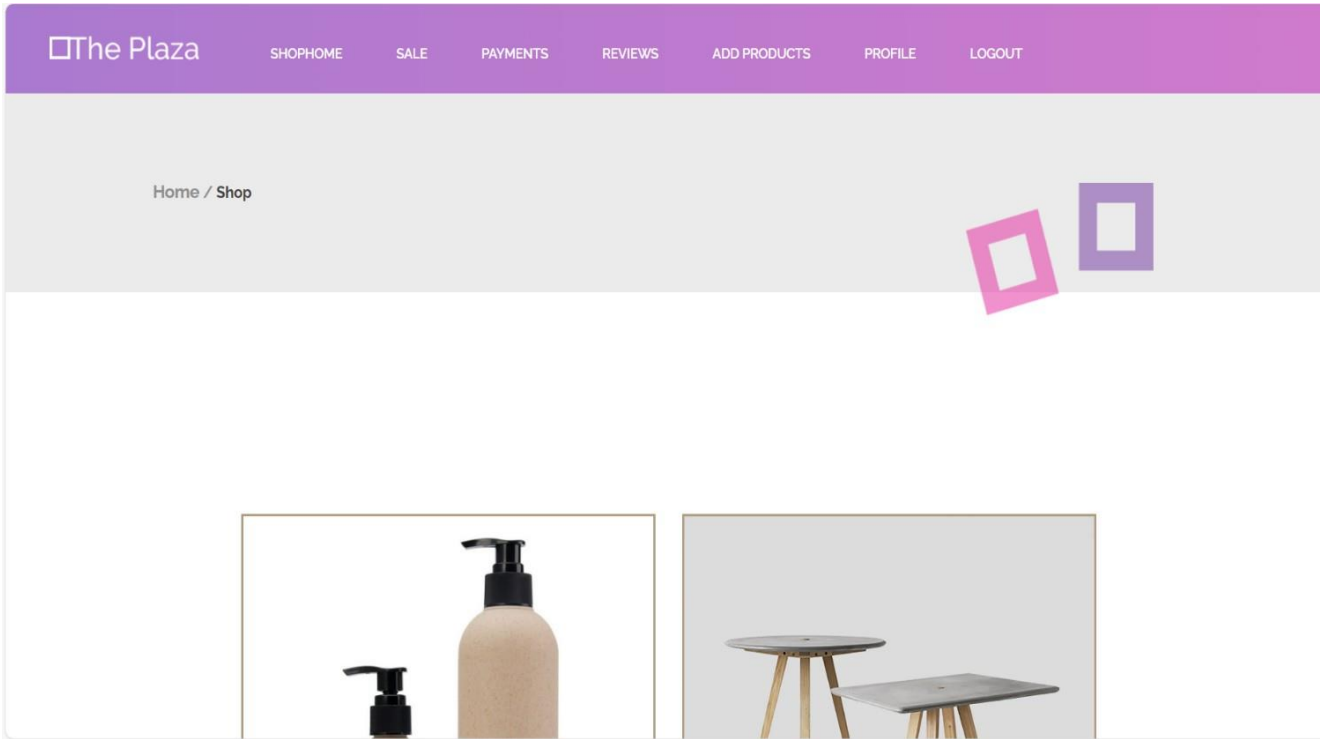
Make Payment

Payment Successful
Image





Payment Successful



Thank you for your payment!

SELLER HOME:



MANAGE PRODUCTS

| Sl No. | Product | Description | Price | |
|--------|---|--|-------|-------------------------------------|
| 1 |  gift for loved ones. | give the best for the best ones! | 1500 | <div>Delete</div> <div>Update</div> |
| 2 |  grooming kit | groom out | 700 | <div>Delete</div> <div>Update</div> |
| 3 |  cultery | try out our new bamboo cultery | 900 | <div>Delete</div> <div>Update</div> |
| 4 |  | y sacrifice Ur health using plastic's try our eco-friendly home intrested | 1999 | <div>Delete</div> |

| | | | |
|---|--|---|------|
| | category | | |
| 4 |  home utensils | y sacrifice Ur health using plastic's try our eco-friendly home utensils! | 1999 |
| 5 |  hair tools | Hair Toooooolsss for uuuu!!!!!!! | 570 |

Product

Image

Price

Description

gift for loved ones.

Choose File

No file chosen

1500

give the best for the best ones!

UPDATE PRODUCT

SAMPLE CODE:

```

from django.db import migrations, models
import django.db.models.deletion

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='buyer',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('firstname', models.CharField(max_length=100)),
                ('lastname', models.CharField(max_length=100)),
                ('district', models.CharField(max_length=100)),
                ('place', models.CharField(max_length=100)),
                ('post', models.CharField(max_length=100)),
                ('pincode', models.CharField(max_length=100)),
                ('email', models.CharField(max_length=100)),
                ('phone', models.CharField(max_length=10)),
                ('status', models.CharField(max_length=100)),
            ],
        ),
        migrations.CreateModel(
            name='login',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('username', models.CharField(max_length=100)),
                ('password', models.CharField(max_length=100)),
                ('usertype', models.CharField(max_length=100)),
            ],
        ),
        migrations.CreateModel(
            name='ordermain',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('date_time', models.DateTimeField()),
                ('total', models.IntegerField()),
                ('status', models.CharField(max_length=100)),
            ],
        ),
    ]

```

```

        ('buyer', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.buyer')),
    ],
),
migrations.CreateModel(
    name='products',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('pname', models.CharField(max_length=100)),
        ('details', models.CharField(max_length=100)),
        ('price', models.CharField(max_length=100)),
        ('image', models.ImageField(upload_to='static/uploads')),
    ],
),
migrations.CreateModel(
    name='shop',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('shopname', models.CharField(max_length=100)),
        ('place', models.CharField(max_length=100)),
        ('landmark', models.CharField(max_length=100)),
        ('phone', models.CharField(max_length=100)),
        ('email', models.CharField(max_length=100)),
        ('status', models.CharField(max_length=100)),
        ('login', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.login')),
    ],
),
migrations.CreateModel(
    name='review',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('review', models.CharField(max_length=1000)),
        ('date', models.DateField()),
        ('buyer', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.buyer')),
        ('product',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='e_app.products')),
    ],
),
migrations.AddField(
    model_name='products',
    name='shop',
    field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.shop'),
),

```



```

migrations.CreateModel(
    name='payment',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('amount', models.IntegerField()),
        ('date', models.DateField()),
        ('buyer', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.buyer')),
        ('ordermain',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='e_app.ordermain')),
    ],
),
migrations.CreateModel(
    name='ordersub',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('amount', models.IntegerField()),
        ('quantity', models.IntegerField()),
        ('ordermain',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='e_app.ordermain')),
        ('product',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='e_app.products')),
    ],
),
migrations.AddField(
    model_name='ordermain',
    name='shop',
    field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.shop'),
),
migrations.CreateModel(
    name='complaint',
    fields=[
        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('complaint', models.CharField(max_length=100)),
        ('reply', models.CharField(max_length=100)),
        ('date', models.CharField(max_length=100)),
        ('buyer', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.buyer')),
        ('shop', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.shop')),
    ],
),
migrations.CreateModel(
    name='cart',
    fields=[

```

```

        ('id', models.BigAutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
        ('quantity', models.IntegerField()),
        ('total', models.DecimalField(blank=True, decimal_places=2,
max_digits=10, null=True)),
        ('buyer', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.buyer')),
        ('product',
models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='e_app.products')),
        ('shop', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.shop')),
    ],
),
migrations.AddField(
    model_name='buyer',
    name='login',
    field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
to='e_app.login'),
),
]

```

8. BIBILOGRAPHY

REFERENCES

1. Youtube
 - <https://youtu.be/PgAZ8KzfhO8> - Easy Tutorials
 - <https://www.w3schools.com/> - W3Schools
2. W3Schools

