



Eduvos
Your Education. Your Future.

Assignment Cover Sheet

| | | | |
|-----------------|--|--------------|--|
| Campus | | | |
| Module code | | | |
| Student surname | | Student name | |
| Student number | | | |

Declaration

"I declare that this assignment is my own original work except for source material explicitly acknowledged, and that the same or related material has not been previously, or is being simultaneously, submitted for this or any other course. I also acknowledge that I am aware of the Institution's policy and regulations on honesty in academic work as set out in the Eduvos *Conditions of Enrolment*, and of the disciplinary guidelines applicable to breaches of such policy and regulations."

| |
|-------------------|
| Student signature |
| Date |

Marker Comments

| | | | |
|-------------|--|------------------------|---|
| Marker name | | Internal marker result | % |
| Marker name | | External marker result | % |

Contents

| | |
|--|----|
| Declaration..... | 1 |
| Marker Comments | 1 |
| Introduction | 3 |
| Needs/Problem | 3 |
| Goals/Objectives..... | 4 |
| Class Responsibility Collaborator (CRC) cards..... | 4 |
| Enhanced Entity Relationship Diagram (EERD) | 4 |
| Context Diagram | 5 |
| Data Flow Diagram (DFD) | 6 |
| Use Case Diagram | 7 |
| Coding | 8 |
| Platform..... | 8 |
| PHP..... | 9 |
| HTML | 11 |
| JavaScript | 15 |
| CSS..... | 17 |
| MySQL Table..... | 19 |
| Conclusion | 19 |

Introduction

DragonStone is an e-commerce platform designed to sell sustainable home products online. The platform allows customers to browse products, add items to their cart, place orders, and make payments. Additionally, an Admin Portal allows internal staff to manage products, users, orders, inventory, and system reports with role-based access permissions

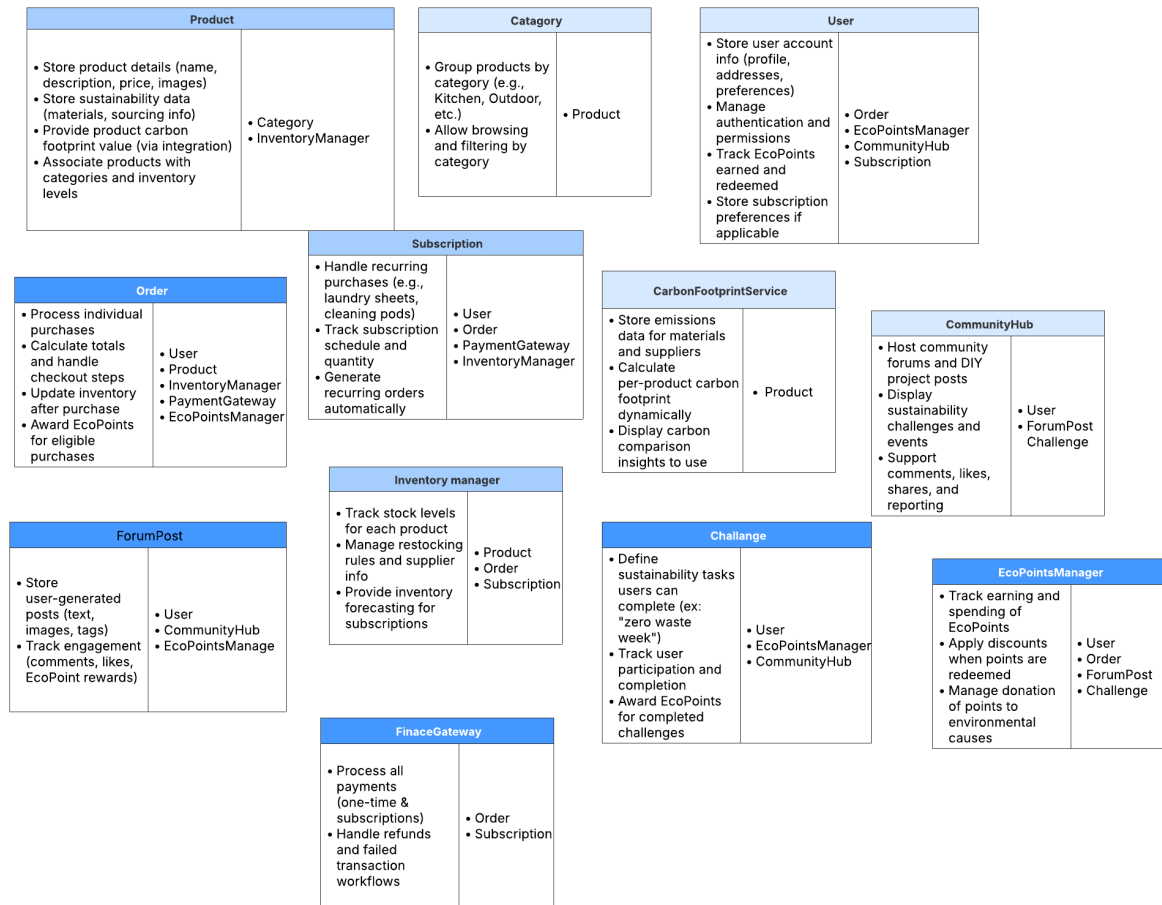
Needs/Problem

DragonStone needs a custom-built platform that consists of two key components: a main customer-facing e-commerce website and an admin management portal. The customer platform must allow users to browse sustainable products, add items to the cart, complete purchases, and subscribe to recurring orders for frequently used household essentials. It should also provide clear product information and a smooth, user-friendly shopping experience.

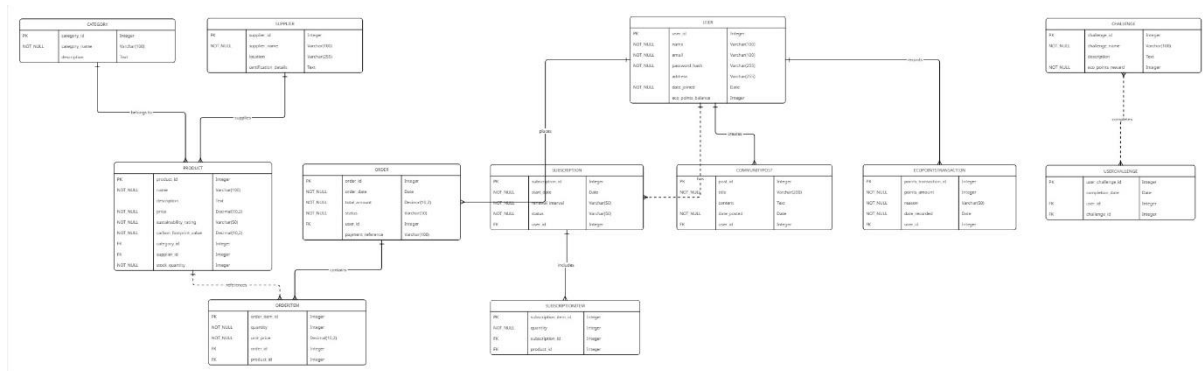
On the operational side, the organisation needs an admin portal where staff can manage products, track inventory, process orders, and oversee user accounts. This platform must include role-based access control, ensuring that employees only have access to features related to their responsibilities.

Goals/Objectives

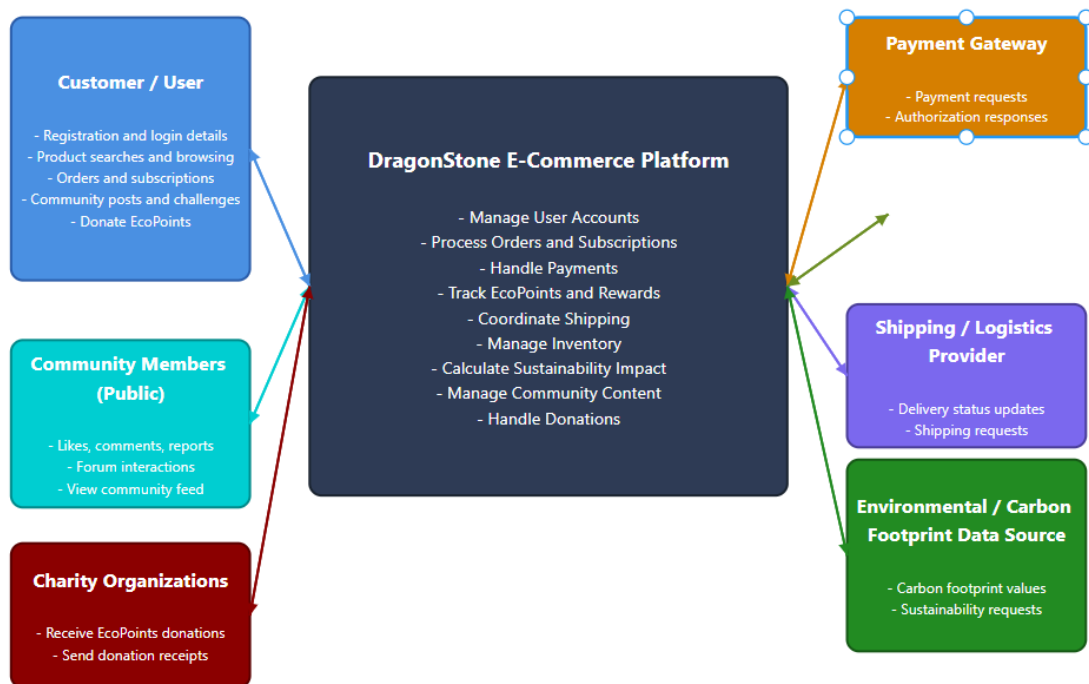
Class Responsibility Collaborator (CRC) cards



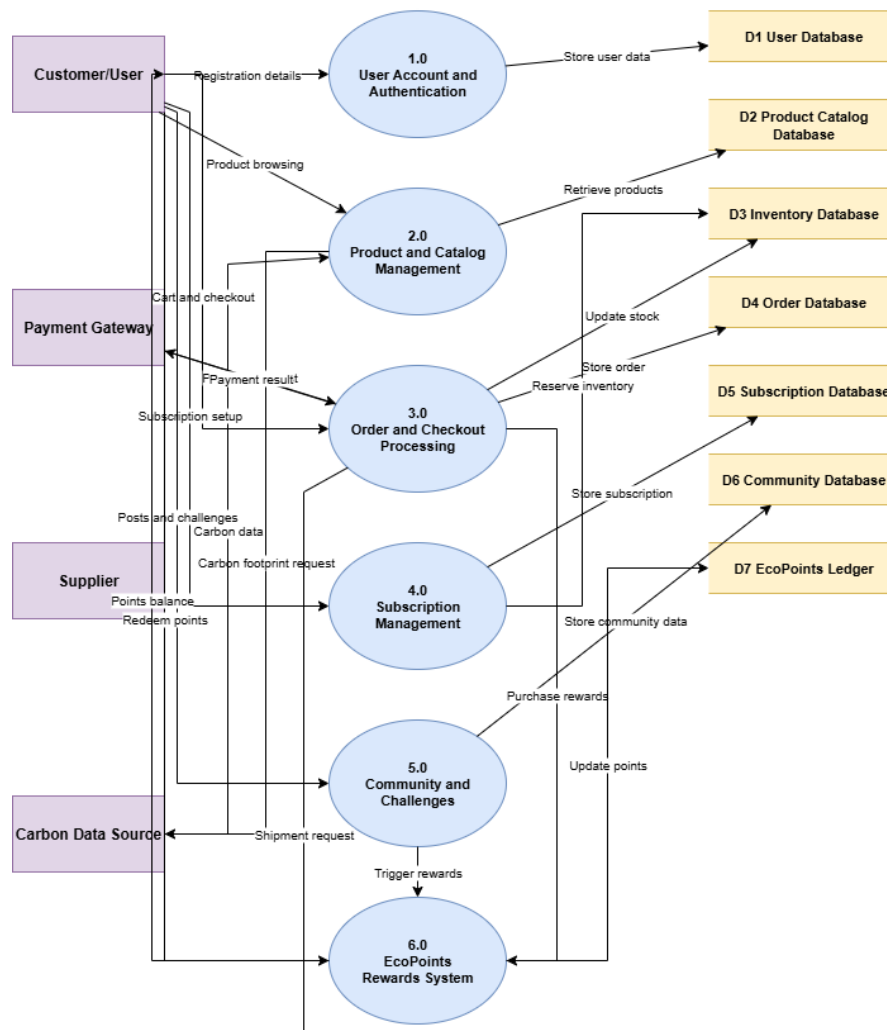
Enhanced Entity Relationship Diagram (EERD)



Context Diagram



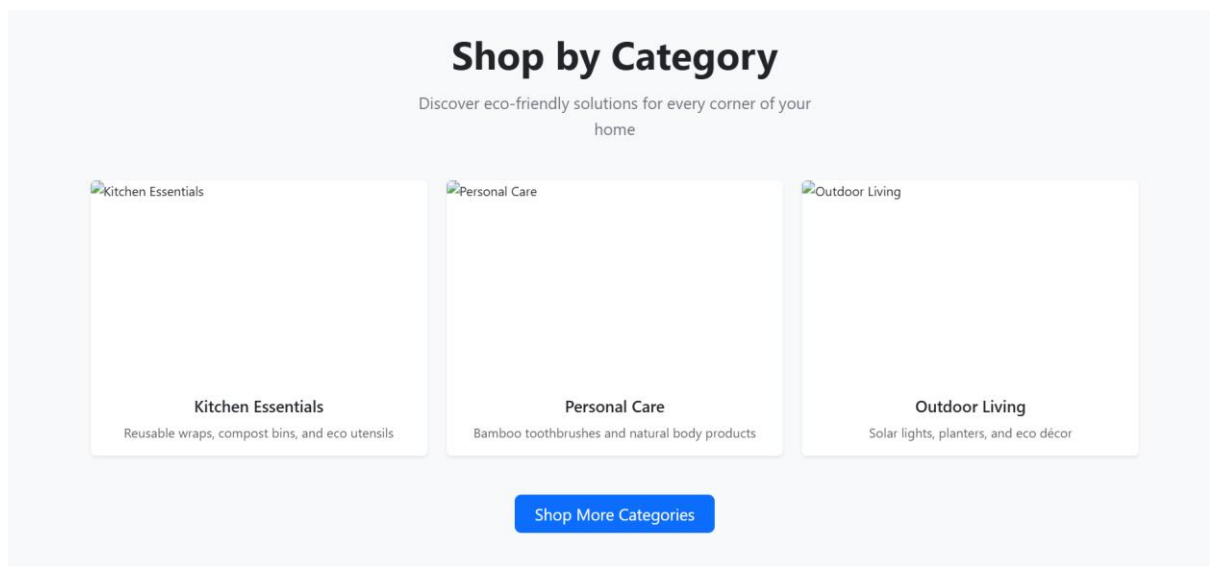
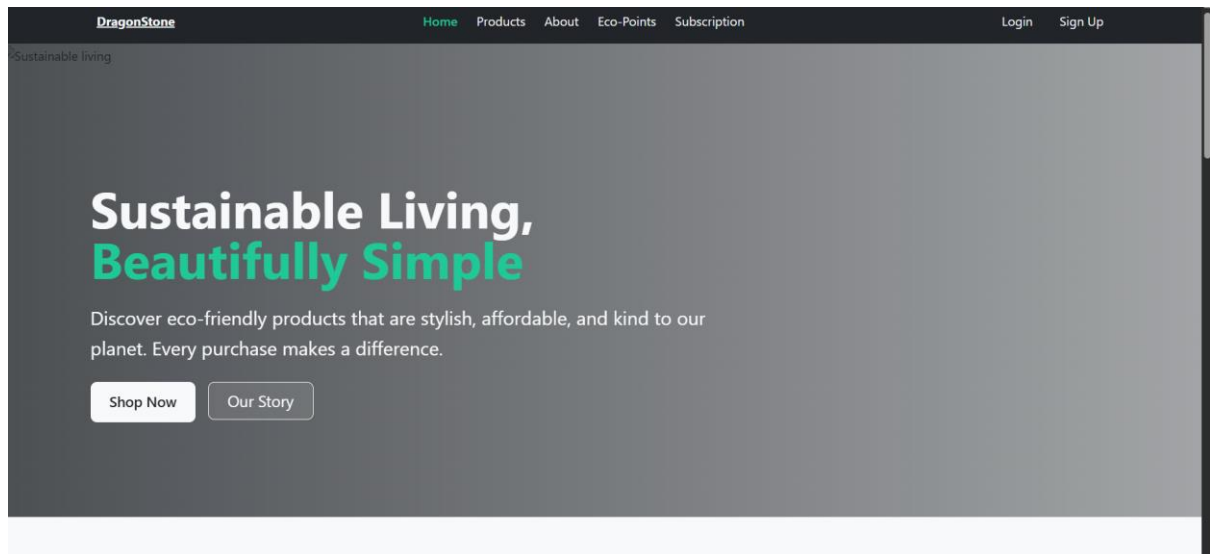
Data Flow Diagram (DFD)

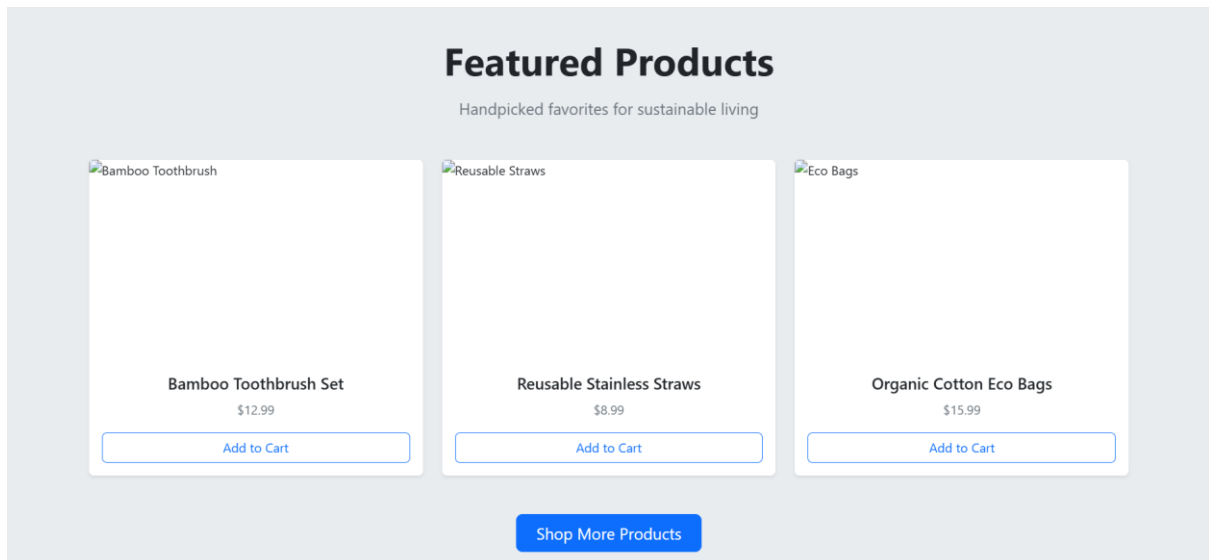


Use Case Diagram



Coding Platform





PHP

```
<?php
```

```
include('background_db_connector.php');
```

```
session_start();
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
```

```
    $FirstNameStr = $_POST['name'] ?? "null";
```

```
    $SurnameStr = $_POST['surname'] ?? "null";
```

```
    $EmailAddressStr = $_POST['email'] ?? "null";
```

```
    $PasswordStr = $_POST['password'] ?? "null";
```

```
    $ConfirmPasswordStr = $_POST['cpassword'] ?? "null";
```

```
    $EmailPatternStr = '/^([a-zA-Z0-9_\.\\-\\+])+\@((([a-zA-Z0-9\\-])+\.)+([a-zA-Z0-9]{2,4})+$)';
```

```
    $PasswordPatternStr = '/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$$%^&*()_\\-+={}[\]|:;,.<?>V]).{8,}$/i';
```

```
    $PasswordMatchBool = $PasswordStr === $ConfirmPasswordStr;
```

```
    if (!preg_match($PasswordPatternStr, $PasswordStr) || !preg_match($EmailPatternStr, $EmailAddressStr) || !$PasswordMatchBool) {
```

```
        $ResultArr = [
```

```
            'success' => false,
```

```
            'duplicate_email' => false,
```

```
            'message' => "Error processing your request",
```

```
];
echo json_encode($ResultArr);
exit();
}
```

```
$HashedPassword = password_hash($PasswordStr, PASSWORD_DEFAULT);
```

```
// $CharactersStr =
'0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
// $CharactersLengthInt = strlen($CharactersStr);
// $PasscodeStr = "";
```

```
// for ($i = 0; $i < 6; $i++) {
//   $PasscodeStr .= $CharactersStr [random_int(0, $CharactersLengthInt - 1)];
// }
```

```
$SQLStr = 'INSERT INTO User (Name, Surname, Email, Password) VALUES (?, ?, ?, ?)';
$stmtObj = $DbConnectionObj->prepare($SQLStr);
$stmtObj->bind_param('ssss', $FirstNameStr, $SurnameStr, $EmailAddressStr,
$HashedPassword);
```

```
try {
    $stmtObj->execute();
```

```
    $ResultArr = [
        'success' => true,
        'message' => "account has been created"
    ];
```

```
    echo json_encode($ResultArr);
} catch (mysqli_sql_exception $e) {
```

```
    switch ($e->getCode()) {
```

```
        case 1062:
```

```
            $ResultArr = [
                'success' => false,
```

```

        'duplicate_email' => true,
        'message' => "Email address already exists"
    ];
    break;
default:
    $ResultArr = [
        'success' => false,
        'duplicate_email' => false,
        'message' => $e->getMessage()
    ];
    break;
}
echo json_encode($ResultArr);
}
}
?>

```

HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DragonStone - Login & Sign Up</title>
<style>
    body {
        margin: 0;
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        background-color: #f4f7f5;
        color: #333;
        display: flex;
        justify-content: center;
    }

```

```
    align-items: center;
    height: 100vh;
}

.auth-container {
    background: #fff;
    padding: 30px;
    border-radius: 12px;
    box-shadow: 0 6px 15px rgba(0,0,0,0.1);
    width: 350px;
}

.auth-container h2 {
    text-align: center;
    margin-bottom: 20px;
    color: #1b4332;
}

.auth-container form {
    display: flex;
    flex-direction: column;
}

.auth-container input[type="text"],
.auth-container input[type="email"],
.auth-container input[type="password"] {
    padding: 12px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 8px;
}

.auth-container button {
    padding: 12px;
    margin-top: 15px;
    background: #40916c;
    border: none;
```

```
    color: #fff;
    border-radius: 8px;
    cursor: pointer;
    font-weight: 600;
}
.auth-container button:hover {
    background: #2d6a4f;
}
.toggle-link {
    text-align: center;
    margin-top: 15px;
}
.toggle-link a {
    color: #1b4332;
    text-decoration: none;
    font-weight: 600;
}
.toggle-link a:hover {
    text-decoration: underline;
}
</style>
</head>
<body>
<div class="auth-container" id="loginForm">
    <h2>Login</h2>
    <form>
        <input type="email" placeholder="Email" required>
        <input type="password" placeholder="Password" required>
        <button id="login">Login</button>
    </form>
    <div class="toggle-link">
        <p>Don't have an account? <a href="#" onclick="toggleForms()">Sign Up</a></p>
```

</div>

</div>

<div class="auth-container" id="signupForm" style="display:none;">

<h2>Sign Up</h2>

<form>

<input type="text" id="name" placeholder="Name" required>

<input type="text" id="surname" placeholder="Surname" required>

<input type="email" id="email" placeholder="Email" required>

<input type="password" id="password" placeholder="Password" required>

<input type="password" id="cpassword" placeholder="Confirm Password" required>

<button id="signup" type="button">Sign Up</button>

</form>

<div class="toggle-link">

<p>Already have an account? Login</p>

</div>

</div>

<script>

function toggleForms() {

const loginForm = document.getElementById('loginForm');

const signupForm = document.getElementById('signupForm');

if (loginForm.style.display === 'none') {

loginForm.style.display = 'block';

signupForm.style.display = 'none';

} else {

loginForm.style.display = 'none';

signupForm.style.display = 'block';

}

}

</script>

</body>

</html>

JavaScript

```
<script>

function toggleForms() {
    const loginForm = document.getElementById('loginForm');
    const signupForm = document.getElementById('signupForm');
    if (loginForm.style.display === 'none') {
        loginForm.style.display = 'block';
        signupForm.style.display = 'none';
    } else {
        loginForm.style.display = 'none';
        signupForm.style.display = 'block';
    }
}

const btnlogin = document.getElementById('login');
const btnsignup = document.getElementById('signup');
const name = document.getElementById('name');
const surname = document.getElementById('surname');
const email = document.getElementById('email');
const password = document.getElementById('password');
const cpassword = document.getElementById('cpassword');

btnsignup.onclick = function () {
    const data = new URLSearchParams();
    data.append('name', name.value);
    data.append('surname', surname.value);
    data.append('email', email.value);
    data.append('password', password.value);
    data.append('cpassword', cpassword.value);
```

```

fetch('../register_account.php',{
    method: 'POST',
    body: data
})
.then((response) => response.json())
.then((result) => {
    alert(result.message);
})
.catch((error) => {

    console.log("error: " + error)
})
}

```

```

btnlogin.onclick = function () {
    const data = new URLSearchParams();
    data.append('email', email.value);
    data.append('password', password.value);
    fetch('../loginAuth.php',{
        method: 'POST',
        body: data
    })
    .then((response) => response.json())
    .then((result) => {
        alert(result.message);
        if(result.success){
            window.location.href = "../Home/Homepage.php";
        }

    })
    .catch((error) => {

```



```
        console.log("error: " + error)
    })
}
```

</script>

CSS

<style>

```
body {
    margin: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: #f4f7f5;
    color: #333;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.auth-container {
    background: #fff;
    padding: 30px;
    border-radius: 12px;
    box-shadow: 0 6px 15px rgba(0,0,0,0.1);
    width: 350px;
}

.auth-container h2 {
    text-align: center;
    margin-bottom: 20px;
    color: #1b4332;
}

.auth-container form {
    display: flex;
```

```
    flex-direction: column;
}
.auth-container input[type="text"],
.auth-container input[type="email"],
.auth-container input[type="password"] {
    padding: 12px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 8px;
}
.auth-container button {
    padding: 12px;
    margin-top: 15px;
    background: #40916c;
    border: none;
    color: #fff;
    border-radius: 8px;
    cursor: pointer;
    font-weight: 600;
}
.auth-container button:hover {
    background: #2d6a4f;
}
.toggle-link {
    text-align: center;
    margin-top: 15px;
}
.toggle-link a {
    color: #1b4332;
    text-decoration: none;
    font-weight: 600;
}
```

```
.toggle-link a:hover {
  text-decoration: underline;
}
```

</style>

MySQL Table

[illegible]

Conclusion

The DragonStone platform provides a smooth online shopping experience for customers while offering a secure admin system for managing products, users, orders, and inventory. With role-based access and a structured database design, the system supports efficient operations and future scalability. Overall, the platform meets DragonStone's needs for both customer engagement and internal management