

Ambessa Coffee Shop

Virginia Anyim

Sammanfattning:

Min kaffe e-handelssida är en SQL-baserad webbplats som använder SQL, HTML, CSS, PHP och JavaScript. Syftet är att呈现出 olika kaffesorter och ha olika standardfunktioner såsom kundvagn, betalningsformulär, inloggning och registrering för kunder, nyhetsprenumeration samt utskick av e-post med bilagor till vald email via ett kontaktformulär. För att dessa funktioner ska fungera hämtas inmatad och sedan lagrad information från en skapad databas med SQL. Detta i kombination med en modern och simpel layout på webbplatsen skapar en komplett e-handelssida för kaffe. Syftet med detta projekt var att lära mig och utöva befintliga kunskaper från backend-sidan med PHP, för att bygga en dynamisk webbplats som hanterar användardata på ett strukturerat och säkert sätt. Med detta projekt av e-handelssidan har jag insett innebördens av att kombinera frontend och backend för att skapa en bra interaktiv användarupplevelse och en komplett webbplats.

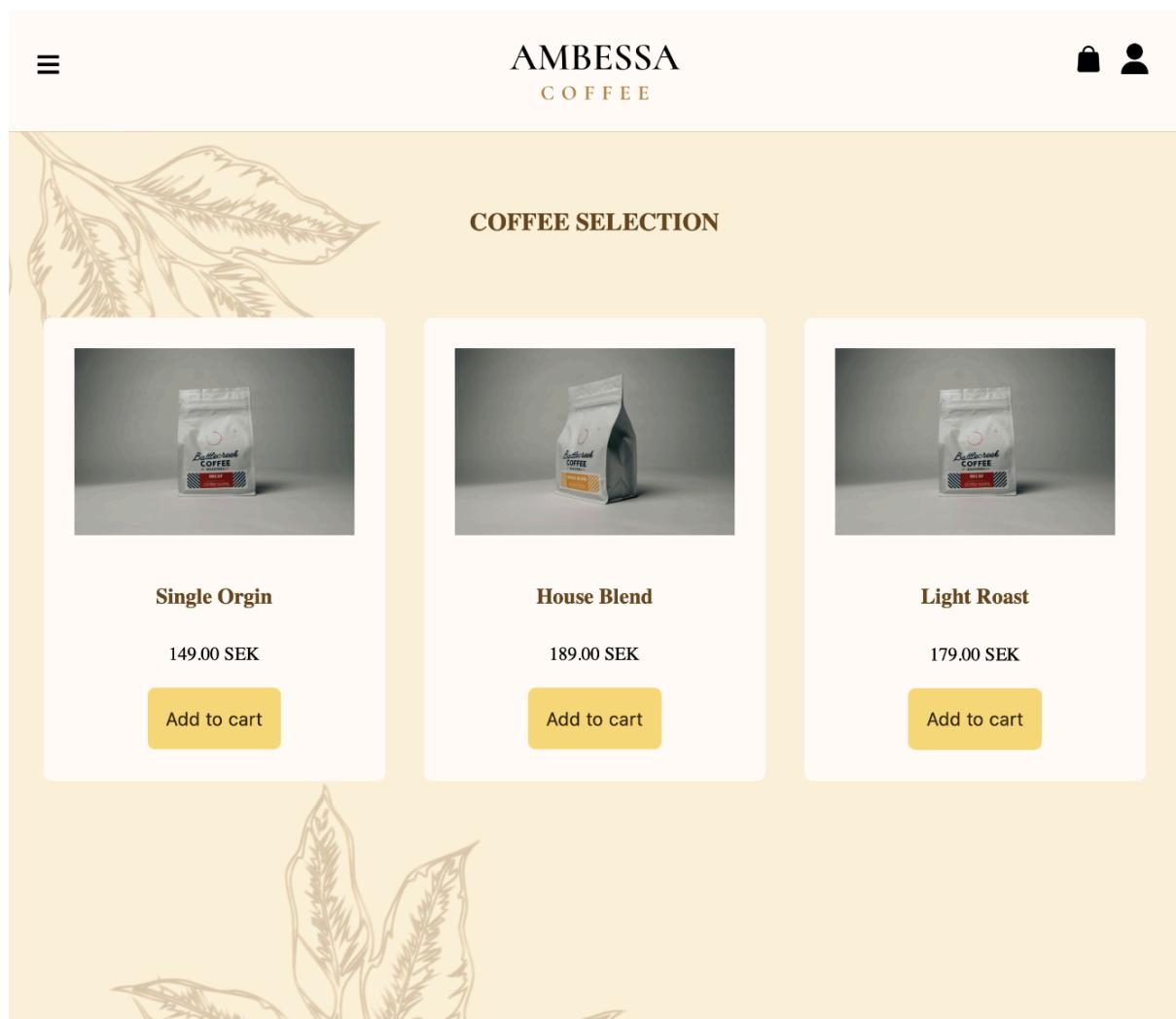


Bild 1: Helhetsbild på webbplatsen

Framtagande:

Det första steget var att brainstorma idéer om vad för typ av e-affär jag ville göra. Sedan kom jag på att min pojkvän ville starta ett kaffeföretag och därför tänkte jag att det skulle vara passande att göra en e-handelssida för kaffeförsäljning. Med denna ide formulering började jag tänka ut vilken känsla denna e-handel skulle förmedla och började då att titta på olika designförslag på google, för just kaffeföretag. Jag insåg snabbt att jag vill använda mig av jordnära samt bruna färger som lätt kan förknippas med kaffe. Dels tittade jag på denna inspirationsbild och fick inspiration för hur min framsida kunde se ut (se bild 2 och 3).

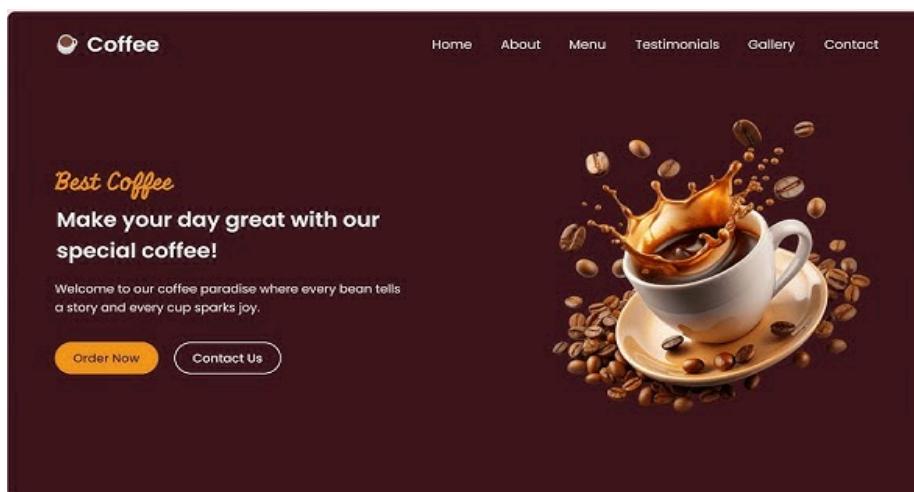


Bild 2: Inspirationsbild för startsidan

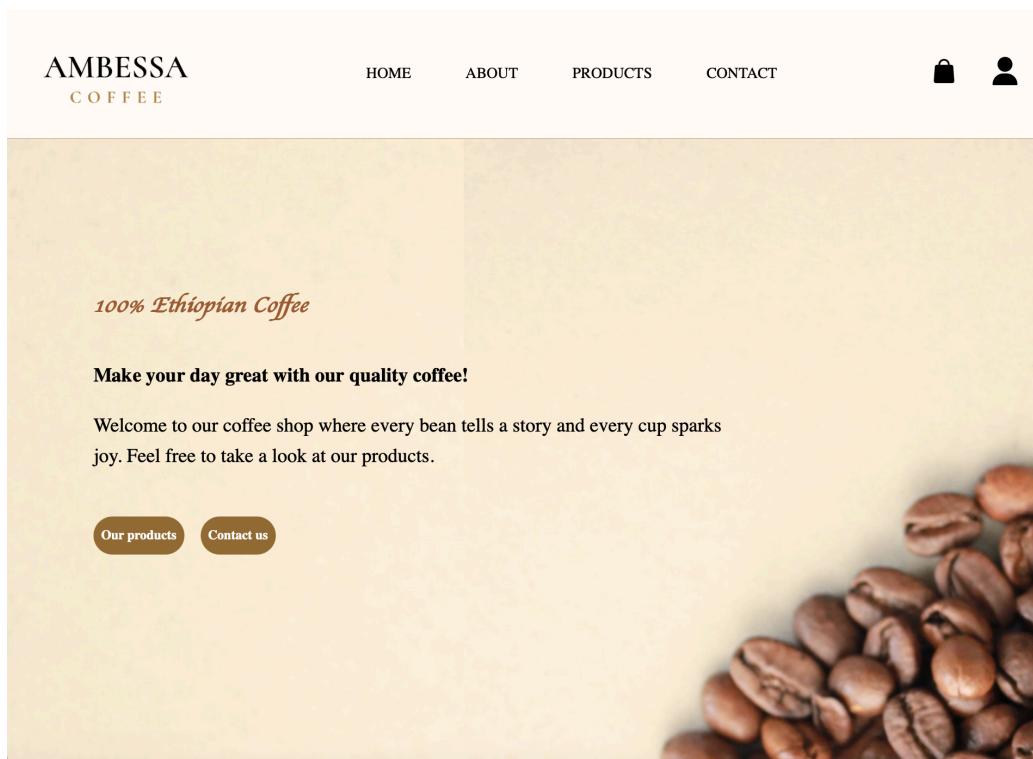


Bild 3: Startsidan för min e-handelssida

Steg 2 var att börja fundera och bestämma över vilka funktioner denna e-handelssida skulle ha. För att göra det tog jag inspiration från exemplena givna i uppgiftsbeskrivningen men jag komunicerade även med min pojkvän angående vilka krav han har på denna sida, för att denna kaffewebbplats kan komma till bruk för honom i framtiden. För mig låg det som prioritering att ha sessionshantering samt kaka för automatisk identifiering. Detta var viktigt att ha i en e-handelssida då php-kod för inloggning, kassa osv lättare kan manipuleras om datorn vet om en specifik session är pågående. Dessutom behövdes standard funktioner som en shoppingvagn, kassa, epostlista, login samt register sida med sparade kunduppgifter i min databas.

Nästa steg var att skapa SQL-databasen via “localhost/phpmyadmin”. För min e-handelssida behövdes följande tabeller under databasen “ambessa_store”: products, subscribers, users, orders. Products innehåller: id, name, price, image, description. Subscribers tabellen innehåller: id, email, subscribed_at och users-tabellen innehåller: id, firstName, lastName, email, password och created_at. Order tabellen innehåller följande rader: id, user_id, customer_name, customer_email, customer_phone, customer_address, payment_method, order_date. För att skapa tabellerna använde jag mig av SQL funktionen, där SQL-kod skrivs in och tabeller skapas automatiskt. Denna kompletta databas möjliggör flera av de funktioner som behövdes för att spara och manipulera användarinformation för en bra användarupplevelse.

I samband med skapandet av SQL-databasen började jag även skapa php-kod för att skriva klart koden som behövdes för samtliga funktioner jag behövde på webbplatsen, men jag insåg snabbt att det var smidigare att fixa html och css koden först innan funktionerna lades på. Jag skrev därför klart all html-kod, dvs alla olika html-sidor samt css-koden för att fixa en snygg layout och passande färger, bilder osv. Tack vare förra kursen “Webbutveckling I” tog det endast ett par dagar att fixa klart denna frontend kod, men layouten med css gjorde att det dröjde lite längre än vad jag hade förutspått då jag först byggde webbplatsen från en dator-enhets vy men kom sedan på att det är bättre att börja från en mobil-enhetsstorlek och skapa media queries för större skärmar då mobiler används främst nu för tiden. Att slutföra html/css först innan appliceringen av backend funktioner funkade bättre för mig då jag även kunde se om eventuella andra funktioner skulle passa att ha på webbplatsen. När allt detta var klart gick jag vidare till steg 4.

Steg 4 innebar att implementera PHP samt JavaScript för att skapa interaktiva gränssnitt. Jag började med att skriva JavaScript-koden för hamburgermenyn, vilket var en enkel men viktig funktion för att sidan skulle bli mobilanpassad och användarvänlig. Därefter fortsatte jag med att skriva PHP-koden för login-systemet, där jag använde sessionshantering och cookies för att spara användarens inloggningstillstånd, samt kod för att skicka data till databasen och hämta information från databasen. Jag skrev även PHP-kod för register-sidan så att nya användare kunde skapa konton och få sina uppgifter sparade i databasen. Sedan gjorde jag klart koden för varukorgen samt kassa-sidan där användaren kunde lägga till och ta bort produkter samt genomföra köp. Jag implementerade även en e-postlista där besökare kunde skriva in sin mejladress för att få nyheter och erbjudanden, vilket direkt kopplades till min

“subscribers”-tabell i databasen. JavaScript användes också för att hantera vissa dynamiska delar på sidan som exempelvis uppdatering av varukorgens innehåll utan att ladda om sidan.

Steg 5 var att testa och se helheten, korrigera och ändra layout tills allt kändes komplett. Jag gick igenom alla funktioner på sidan flera gånger för att se att allt fungerade som det skulle, både frontend och backend. Jag testade även sidan från både dator och mobil-vy för att försäkra mig om att layouten anpassade sig korrekt. Vissa delar i designen fick justeras, bland annat paddings, fontstorlekar och bildstorlekar så att sidan blev mer responsiv och estetiskt tilltalande. Jag jämförde även med andra liknande kaffeföretags sidor för att se om jag missat några viktiga funktioner. Då såg jag till exempel att vissa hade filter för olika kaffesorter, vilket jag funderade på att lägga till som en framtida förbättring. När jag kände mig nöjd med både design, funktioner och användarvänlighet så ansåg jag att projektet var färdigt.

Form:

HTML koden i min skapade e-handelssida hanterar innehållet som exempelvis rubriker, texter, bilder, listor, formulär etc. Vidare gör min CSS kod designen av alla dessa element för att skapa en snygg design samt anpassat för målgruppen. Javascript-koden möjliggör de interaktiva och dynamiska beteenden, exempelvis vad som sker vid tryck av knappar. Php-koden hanterar logiken bakom kulisserna, exempelvis så att ett mail från formuläret faktiskt kommer fram till vald mail.

E-handelssidan är uppdelad i flera olika centrala delar som header, main, nav, section och footer. Dessa sektioner delar inte endast upp koden snyggt till utseendet för användaren, utan gör även koden lättare att korrigera och läsa igenom. Vidare i CSS är koden uppdelat i display: grid samt flexbox för att underlätta den responsiva layouten, se exempel nedan. Jag valde att experimentera lite blandat med flex och grid för att se hur de olika fungerar då jag tidigare endast känt till grid.

```
301 .products-grid {  
302   display: grid;  
303   grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));  
304   gap: 20px;  
305   margin-top: 30px;  
306 }  
307  
483  
484 .cart-item {  
485   display: flex;  
486   align-items: flex-start;  
487   gap: 1rem;  
488   margin-bottom: 1.5rem;  
489   border-bottom: 1px solid #ddd;  
490   padding-bottom: 1rem;  
491 }  
492  
493 .cart-details {  
494   flex: 1;  
495 }
```

Bild 4: Användning av flex och grid

För att koppla min kod till den skapade databasen “ambessa_store” med de olika tabellerna behövdes koden nedan i db.php, se bild 5 nedan.

```
1 <?php
2 //Kopplingsinställningar till databasen
3 $host = 'localhost';
4 $db = 'ambessa_store';
5 $user = 'root';
6 $pass = '';
7
8 //Ansluter till databasen
9 $conn = new mysqli(hostname: $host, username: $user, password: $pass, database: $db);
10
11 //Kontroll av anslutningen
12 if ($conn->connect_error) {
13     echo("Connection fail: " . $conn->connect_error);
14 }
15
16 
```

Bild 5: Koppling till databasen

Dessutom ville jag ha sessioner på viktiga sidor av min webbplats som login, register, cart, checkout osv och därför skapades start_session.php som sedan implementeras i valda php-filer med funktionen require_once 'start_session.php'. När session_start() metoden körs i php försöker den återanvända en befintlig session om webbläsaren skickar med en giltig session_cookie. Om sådan cookie finns återupptas samma session och sessionens data \$_SESSION blir tillgänglig och används på sidan. Om ingen session-cookie finns, exempelvis om det är användarens första besök eller om en utloggning har skett, skapas en ny session och cookie skickas till webbläsare. Detta möjliggör exempelvis att kundkorgen rensas vid utloggning eller återupptas vid inloggning.

```
1 <?php
2 //starta session eller fortsätter en påbörjad session
3 session_start();
4
5 //Kontroll om session-id redan finns, om inte skapa nytt id för sessionen
6 if (!isset($_SESSION['session_id'])) {
7     session_regenerate_id(delete_old_session: true);
8     //sparar nya id:t i sessionen
9     $_SESSION['session_id'] = session_id();
10 }
11
12 //Kollar om användaren har en cookie för session id
13 if (!isset($_COOKIE['session_id'])) {
14     //skapa en cookie med session id som varar i 1 timme
15     setcookie(name: "session_id", value: $_SESSION['session_id'], expires_or_options: time() + 3600, path: "/");
16 }
17 
```

Bild 6: Sessions och cookie-hantering

I navigeringsmenyn skapade jag en template med id= "mini-cart-item-template" samt en tillhörande div-class för lättare manipulering av tillhörande design. Denna har i uppgift att

verka som en dold mall för kundvagnsobjekt, texten som står i html dokumentet nedan har i uppgift att vara placeholders som sedan ersätts med aktuell data (se bild 7 och 8).

```
10 <body id = "home">
11   <header>
12     <div class="header-container">
13       <div class="menu-toggle">&#9776;</div>
14
15       <a href="/uppgifter/9/html/index.html" class="logo-link">
16         
17       </a>
18
19     <nav class="navbar">
20       <a href="/uppgifter/9/html/home.html">Home</a>
21       <a href="/uppgifter/9/html/about.html">About</a>
22       <a href="/uppgifter/9/php/products.php">Products</a>
23       <a href="/uppgifter/9/html/contact.html">Contact</a>
24     </nav>
25
26     <div class="icons">
27       <a href="../php/cart.php" class="cart-icon-wrapper">
28         
29       </a>
30       <a href="/uppgifter/9/php/profile-replace.php">
31         
32       </a>
33     </div>
34     <div id="mini-cart" class="mini-cart-dropdown">
35       <h4>Shopping Bag</h4>
36       <div id="cart-items"></div>
37       <div id="cart-total"></div>
38     </div>
39
```

Bild 7:

```
40   <template id="mini-cart-item-template">
41     <div class="mini-cart-item">
42       
43       <div class="cart-item-details">
44         <strong>---name---</strong><br>
45         Antal: ---quantity---<br>
46         Pris: ---price--- kr
47       </div>
48     </div>
49   </template>
50 </div>
51 </header>
```

Bild 8:

Vid klick av användargubben i menyn har användaren chans att logga in eller registrera sig och bäge av dessa är uppbyggda genom ett formulär på html sidorna som sedan dirigerar vidare till en enskild php sida (login.php eller register.php) med hjälp av post-metoden. Jag valde post-metoden vid samtliga formulär på denna webbplats då den bättre och säkrare hanterar känslig information till skillnad från get-metoden. För att möjliggöra att en inloggad användare kommer till sin profil-dashboard och en användare som ej är inloggad istället kommer till login-sidan, behövde jag skapa en separat profile-replace.php som genomför en kontroll om användaren är inloggad och isåfall visar profilens innehåll som email ordrar osv, annars ber användaren att logga in.

```

1  <?php
2  require_once 'start_session.php';
3  require_once 'db.php';
4
5  //Kontroll om användaren är inloggad annars vidare till login sida
6  if (!isset($_SESSION['user_id'])) {
7      header(header: "Location: ../html/login.html");
8      exit();
9  }
10
11 //Hämtar användarens ID från sessionen
12 $userId = $_SESSION['user_id'];
13
14 //Hämta användarens information från databasen
15 $sqlUser = "SELECT firstName, lastName, email FROM users WHERE id = $userId";
16 $result = $conn->query(query: $sqlUser);
17
18 //Kollar om användaren finns i databsen, hämta namn, email, annars stopp med felmeddelande
19 if ($result && $result->num_rows > 0) {
20     $user = $result->fetch_assoc();
21     $fullName = $user['firstName'] . ' ' . $user['lastName'];
22     $email = $user['email'];
23 } else {
24     die("User not found.");
25 }
26
27 //Hämta användarens eventuella orders från databasen
28 $sqlOrders = "SELECT order_details, order_date FROM orders WHERE customer_email = '$email' ORDER BY order_date DESC";
29 $result = $conn->query(query: $sqlOrders);
30
31 $ordersText = "";
32 if ($result && $result->num_rows > 0) {
33     while ($order = $result->fetch_assoc()) {
34         $ordersText .= "Date: " . $order['order_date'] . "\n";
35         $ordersText .= $order['order_details'] . "\n\n";
36     }
37 } else {
38     $ordersText = "You have no orders yet.";
39
40 // Läs HTML-mallen för profilen
41 $html = file_get_contents(filename: '../html/profile.html');
42
43 // Ersätt platshållare med användardata och orders
44 $html = str_replace(search: '---username---', replace: $fullName, subject: $html);
45 $html = str_replace(search: '---user-email---', replace: $email, subject: $html);
46 $html = str_replace(search: '---user-orders---', replace: nl2br(string: htmlspecialchars(string: $ordersText)), subject: $html);
47
48
49 // Visa sidan
50 echo $html;

```

Bild 9: Profile-replace.php

Längst ner i footern finns chansen att gå till olika länkar som exempelvis: FAQ, our story, shipping, returns etc men dessa är tänkta att fyllas med information i framtiden. Det finns även ett litet formulär där användaren kan prenumerera på nyhetsbrev. Den funktionen fungerar med hjälp av post-metoden i formuläret som sänder formuläret till php sidan "subscribe.php".

```
<div class="footer-section">
  <h4>STAY UPDATED</h4>
  <form action="../php/subscribe.php" method="post">
    <input type="email" name="email" placeholder="Email address" required>
    <button type="submit">Subscribe</button>
  </form>
</div>
</div>
```

Bild 10: subscribe.php

Kontaktformuläret är till för om kunden behöver hjälp snabbt och effektivt vill skicka iväg en fråga rörande olika frågor. Det sker via process.php filen och skickas till min nyskapade gmail för detta projekt, tanken är att mailen senare ska ändras till en faktiskt företagsmail. Kontaktformuläret ser ut på följande sätt i html-filen:

```
<main>
  <section id = "form">
    <h1>Contact Us</h1>
    <p>Do you have any questions or want to get in touch? Fill out the form below, we strive to respond within <strong>1-3 business days</strong>.</p>
    <form method="post" enctype="multipart/form-data" action="../php/process.php">
      <fieldset>
        <input type="hidden" name="to" value="virginia.anyim1@gmail.com">
        <label for="name">Name:</label>
        <input type="text" name="name" id="name" required><br><br>
        <label for="email">E-mail:</label>
        <input type="email" name="email" id="email" required><br><br>
        <label for="subject">Subject:</label>
        <select id="subject" name="subject">
          <option value="Order">Order</option>
          <option value="Payment">Payment</option>
          <option value="Return">Return</option>
          <option value="Collaboration">Collaboration</option>
          <option value="Other">Other</option>
        </select>
        <label for="message">Message:</label><br>
        <textarea name="message" id="message" rows="6" required></textarea><br><br>
        <label for="file">Attach file:</label>
        <input type="file" id="file" name="file">
        <input type="submit" class="long-btn" value="Send">
      </fieldset><br>
    </form>
  </section>
</main>
```

Bild 11: contact.html

```

1  <?php
2  // inkluderar PHPMailer-biblioteket och skapar ett nytt PHPMailer-objekt
3  require 'PHPMailerAutoload.php';
4  $mail = new PHPMailer;
5
6  // Kontroll om formuläret skickades via post + hämta formulärets innehåll
7  if ($_SERVER["REQUEST_METHOD"] == "POST") {
8      $to = $_POST['to'];
9      $subject = $_POST['subject'];
10     $message = $_POST['message'];
11     $fromEmail = $_POST['email'];
12     $fromName = $_POST['name'];
13
14     // SMTP-inställningar för att skicka via (gmail)
15     $mail->isSMTP();
16     $mail->Host = 'smtp.gmail.com';
17     $mail->Port = 587;
18     $mail->SMTPAuth = true;
19     $mail->Username = 'virginia.anyim1@gmail.com';
20     $mail->Password = 'rvdr sult nwjm vjuy';
21     $mail->SMTPSecure = 'tls';
22
23     // Sätter avsändare och mottagare
24     $mail->setFrom(address: $fromEmail, name: $fromName);
25     $mail->addAddress(address: $to);
26
27     // Filbilaga läggs till om en fil har laddats upp via formuläret
28     if (isset($_FILES['file']) && $_FILES['file']['error'] == UPLOAD_ERR_OK) {
29         $mail->addAttachment(path: $_FILES['file']['tmp_name'], name: $_FILES['file']['name']);
30     }
31
32     // Ämne och meddelande
33     $mail->Subject = $subject;
34     $mail->Body = $message . "\n\n This message is sent from a contact form!";
35
36     // Skicka e-post, om lyckat skicka användaren tillbaka till formulär annars...
37     if ($mail->send()) {
38         header(header: "Location: ../html/contact.html?success=1");
39         exit();
40     } else {
41         // ...skicka till formulärsidan med error
42         header(header: "Location: ../html/contact.html?error=1");
43         exit();
44     }
45 }

```

Bild 12: process.php

I kundvagnen skapade jag funktioner såsom att kunna minska och öka antalet av valda produkter samt borttagning helt. Mallen hämtas från cart_view.html som använder placeholders för ersättning av riktig data.

```

1  <?php
2  require_once 'start_session.php';
3  require_once 'db.php';
4
5 //Hanterar kundvagnsknapparna, öka, minska och ta bort produkt
6 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
7     $action = isset($_POST['action']) ? $_POST['action'] : '';
8     $productId = isset($_POST['product_id']) ? (int)$_POST['product_id'] : 0;
9
10    //Kontroll att produkt id är giltig och produkten finns i kundvagnen
11    if ($productId > 0 && isset($_SESSION['cart'][$productId])) {
12        if ($action === 'increase') {
13            $_SESSION['cart'][$productId]++;
14        } elseif ($action === 'decrease') {
15            $_SESSION['cart'][$productId]--;
16            //Tar bort produkten om antalet blir 0 eller mindre
17            if ($_SESSION['cart'][$productId] <= 0) {
18                unset($_SESSION['cart'][$productId]);
19            }
20        } elseif ($action === 'remove') {
21            unset($_SESSION['cart'][$productId]);
22        }
23    }
24    //Skicka användaren tillbaka till kundvagnssidan
25    header(header: "Location: cart.php");
26    exit();
27 }
28
29 //Ladda html mallen för kundvagnen
30 $page = file_get_contents(filename: '../html/cart_view.html');

```

```

40 //Om kundvagnen är tom, meddelande
41 if (empty($_SESSION['cart'])) {
42     $cartItemsHtml = "Your cart is empty.";
43     $totalText = "";
44     $checkoutText = "";
45 } else {
46     //Annars skapa en lista med produkt id: för SQL frågan med kommatecknen i mellan
47     $productIds = implode(separator: ',', array: array_keys(array: $_SESSION['cart']));
48     $sql = "SELECT id, name, price, image FROM products WHERE id IN ($productIds)";
49     $result = $conn->query(query: $sql);
50
51     //Loopa genom varje produkt som finns i kundvagnen
52     if ($result && $result->num_rows > 0) {
53         while ($product = $result->fetch_assoc()) {
54             $id = $product['id'];
55             $quantity = $_SESSION['cart'][$id];
56             $subtotal = $product['price'] * $quantity;
57             $totalPrice += $subtotal;
58
59             //Skapar sökväg till produktbilden
60             $imagePath = '../images/' . $product['image'];
61
62             //Byt ut placeholder i produktmallen med riktiga värden (med 2 decimaler)
63             $itemHtml = str_replace(
64                 search: ['---NAME---', '---IMAGE---', '---QUANTITY---', '---PRICE---', '---SUBTOTAL---', '---PRODUCT_ID---'],
65                 replace: [
66                     htmlspecialchars(string: $product['name']),
67                     htmlspecialchars(string: $imagePath),
68                     $quantity,
69                     number_format(num: $product['price'], decimals: 2),
70                     number_format(num: $subtotal, decimals: 2),
71                     $id
72                 ],
73                 subject: $productItemTemplate
74             );

```

```

102    <template id="cart-item-template">
103        <div class="cart-item">
104            <div class="cart-item-inner">
105                
106                <div class="cart-details">
107                    <h2>---NAME---</h2>
108                    <p>Quantity: ---QUANTITY---</p>
109                    <p>Price: ---PRICE--- SEK/each</p>
110                    <p>Subtotal: ---SUBTOTAL--- SEK</p>
111
112                    <form method="post">
113                        <input type="hidden" name="product_id" value="---PRODUCT_ID---" />
114                        <button type="submit" name="action" value="decrease">-</button>
115                    </form>
116
117                    <form method="post">
118                        <input type="hidden" name="product_id" value="---PRODUCT_ID---" />
119                        <button type="submit" name="action" value="increase">+</button>
120                    </form>
121
122                    <form method="post">
123                        <input type="hidden" name="product_id" value="---PRODUCT_ID---" />
124                        <button type="submit" name="action" value="remove">Remove</button>
125                    </form>

```

Jag använde mig även av json för cart_handler.php och cart-status.php som möjliggör addering av produkter i varukorgen i mini-carten.

```

1  <?php
2  require_once 'start_session.php';
3  require_once 'db.php';
4
5 //Svar tillbaka är i json format
6 header(header: 'Content-Type: application/json');
7
8 //Kontroll att det är ett formulär med post
9 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
10
11    //Hämtar variabler från formuläret
12    $action = isset($_POST['action']) ? $_POST['action'] : '';
13    $productId = isset($_POST['product_id']) ? (int)$_POST['product_id'] : 0;
14    $quantity = isset($_POST['quantity']) ? (int)$_POST['quantity'] : 1;
15
16    //Om användaren vill lägga till i kundvagnen och id är giltigt
17    if ($action === 'add' && $productId > 0) {
18
19        //Om kundvagnen inte finns änna skapa den
20        if (!isset($_SESSION['cart'])) {
21            $_SESSION['cart'] = array();
22        }
23
24        //Om produkten inte finns i kundvagnen än läggs den till
25        if (!isset($_SESSION['cart'][$productId])) {
26            $_SESSION['cart'][$productId] = 0;
27        }
28
29        //Öka antalet
30        $_SESSION['cart'][$productId] += $quantity;
31
32        //Skicka tillbaka ett svar
33        echo json_encode(value: array('success' => true));
34        exit;
35    }
36
37    //Om något gick fel skicka felmeddelande
38    echo json_encode(value: array('success' => false, 'message' => 'Error'));
39
40
41 //Svar tillbaka är i json format
42 header(header: 'Content-Type: application/json');
43
44 //Skapar en tom lista för att fylla med produkter från kundvagnen
45 $items = [];
46
47 //Kollar om det finns något i kundvagnen
48 if (!empty($_SESSION['cart'])) {
49
50    //Bygger en sträng med id:n, säkerställs att id:t är av heltal
51    $idString = "";
52    foreach ($_SESSION['cart'] as $productId => $quantity) {
53        $idString .= intval(value: $productId) . ",";
54    }
55
56    //Tar bort sista kommatecknet
57    $idString = rtrim(string: $idString, characters: ",");
58
59    //Hämtar produkterna från databasen
60    $sql = "SELECT id, name, price, image FROM products WHERE id IN ($idString)";
61    $result = $conn->query(query: $sql);
62
63    //Går igenom varje rad och lägger till i kundvagnens lista
64    if ($result) {
65
66        while ($row = $result->fetch_assoc()) {
67            $id = $row['id'];
68            $quantity = $_SESSION['cart'][$id];
69
70            $items[] = [
71                'id'      => $id,
72                'name'   => $row['name'],
73                'price'  => $row['price'],
74                'image'  => $row['image'],
75                'quantity' => $quantity
76            ];
77        }
78    }
79
80 }
81
82 //Skickar tillbaka kundvagnens innehåll som json
83 echo json_encode(value: ['items' => $items]);
84
85

```

Psuedokod på helheten:

Startsidan:

```
start_page_args() = ""

call start_session()
call db_connect()

if current_page == "home" then
    call load_home_page()
else if current_page == "products" then
    call load_products_page()
else if current_page == "contact" then
    call load_contact_page()
else if current_page == "about" then
    call load_about_page()
else if current_page == "cart" then
    call load_cart_page()
else if current_page == "profile" then
    call load_profile_page()

call show_header()
call show_footer()

return "page loaded"
```

Sessionshantering: (start_session.php)

```
start_session_args() = ""

if session_id not exists then
    generate new session_id
    set cookie with session_id, expires 1 hour
    store session_id in server_memory

return "session started"
```

Databasanslutning (db_connect)

```
db_connect_args() = ""
connect to database "ambessa_store" on "localhost"
with username "root" and password ""
```

```

if connection fails then
    show "Database connection error"

return "database connected"

```

Hantering av produkter

```

load_products_page_args() = ""

products = get_products_from_database("SELECT * FROM
products")

for each product in products
    show product.image
    show product.name
    show product.price
    show_button("add to cart.")

return "products page loaded"

```

Hantering av kundvagnen

```

load_cart_page_args() = ""

if cart is empty then
    show "Your cart is empty."
else
    product_ids = get_product_ids_from_cart()
    products = get_products_from_database("SELECT * FROM
products WHERE id IN (product_ids)")

    for each product in products
        show product.image
        show product.name
        show product.price
        show quantity from cart
        show total_price = product.price * quantity
        show_button("Increase quantity")
        show_button("Decrease quantity")
        show_button("Remove product")

    show calculate_cart_total(products)

```

```

show_button("Go to checkout")

if action == "increase" then
    add 1 to product quantity in cart
else if action == "decrease" then
    subtract 1 from product quantity in cart
    if quantity <= 0 then
        remove product from cart
else if action == "remove" then
    remove product from cart

return "cart page loaded"

```

Hantering av användare

```

load_login_page_args() = ""

if user is already logged in then
    call load_profile_page()
else
    show_form("Email input, Password input, Submit button")

if form_submitted then
    email = get_form_data("email")
    password = get_form_data("password")

    user = find_user_in_database("SELECT * FROM users WHERE
email = email")

    if user exists and password matches then
        store_user_session(user.id, email)
        call load_profile_page()
    else
        show "invalid email or password"

return "login page loaded"

load_register_page_args() = ""

show_form("First name, Last name, Email, Password, Confirm
password")

if form_submitted then

```

```

if any field empty then
    show "All fields are required"
    return "registration failed"

if password != confirm_password then
    show "Passwords don't match"
    return "registration failed"

if email_exists_in_database("SELECT id FROM users WHERE
email = email") then
    show "Email already registered"
    return "registration failed"

create_new_user_in_database("INSERT INTO users (firstName,
lastName, email, password) VALUES (...)")
store_user_session(user_info)
call load_profile_page()

return "registration page loaded"

load_profile_page_args() = ""

if user not logged in then
    call load_login_page()

user = get_user_data_from_database("SELECT * FROM users WHERE
id = session_user_id")
orders = get_user_orders_from_database("SELECT * FROM orders
WHERE customer_email = user.email")

show user information
show order history
show_button("Logout")

return "profile page loaded"

```

Hantering av beställning

```

load_checkout_page_args() = ""

if cart is empty then
    call load_products_page()

```

```

if user not logged in then
    call load_login_page()

show_form("Full name, Email, Phone, Shipping address, Payment
method")
show cart summary
show_button("Place order")

return "checkout page loaded"

process_order_args() = ""

if request_method == "POST" then
    if any required field is empty then
        show "All fields are required"
        return "order failed"

    order_id = insert_into_database("INSERT INTO orders (... )
VALUES (...) ")
    send_confirmation_email(order_id)
    clear_session_cart()
    call load_order_confirmation_page(order_id)

return "order processed"

```

Hantering av e-post

```

process_contact_form_args() = ""

if request_method == "POST" then
    if any required field is empty then
        show "All fields are required"
        return "contact failed"

    mail = create_email_object()
    set_mail_properties(mail)
    set_mail_content(mail, form_data)

    if attachment exists then
        add_attachment(mail)

```

```

if send_mail(mail) == true then
    show "Message sent successfully"
else
    show "Error sending message"

return "contact form processed"

process_subscription_args() = ""

if request_method == "POST" then
    email = get_post_data("email")

    if email_exists_in_database("SELECT id FROM subscribers
WHERE email = email") then
        show "Email already subscribed"
        return "subscription failed"

    insert_into_database("INSERT INTO subscribers (email) VALUES
(email)")
    show "Welcome to the team"

return "subscription processed"

```

Funktion:

Headern består av en navigationsmeny som finns på samtliga sidor där användaren kan navigera sig mellan: home, about, products, contact och även två symboler - en kundvagn och en användargubbe för profilen. Jag hade dessutom min egenskapade företagslogga i headern som är klickbar med en omdirigeringslänk till index.html, för att användare snabbare ska komma tillbaka till startsidan.

I startsidan möts användaren av en simpel layout där det finns två knappar (contacts us och products) I menyn går det att navigera sig till home, about, products och contact. Nedan ser vi utseendet av startsidan på en mobil enhet samt utfällda hamburgermenyn som finns tillgänglig i enheter under 768px.

AMBESSA COFFEE

100% Ethiopian Coffee

Make your day great with our quality coffee!

Welcome to our coffee shop where every bean tells a story and every cup sparks joy. Feel free to take a look at our products.

[Our products](#) [Contact us](#)

SHOP

- Shipping
- Returns
- Wholesale

ABOUT

- Our story
- Privacy policy
- Terms and conditions

AMBESSA COFFEE

[Home](#)

[About](#)

[Products](#)

[Contact](#)

Make your day great with our quality coffee!

Går vi vidare till about sidan möts vi av en sida med information om kaffeföretaget.

AMBESSA COFFEE

[HOME](#) [ABOUT](#) [PRODUCTS](#) [CONTACT](#)

About Us

We are Yoel Eyassu and Siem Eyassu — brothers, coffee lovers, and the founders of Ambessa Coffee.

Our Story

At Ambessa Coffee, everything starts with passion. We grew up in Eritrea, where coffee is more than just a drink — it's culture, tradition, and a way to bring people together. From a young age, we were surrounded by the smell, taste, and craft of real coffee.

With hands-on experience in coffee-making and a strong connection to our heritage, we decided to create a company that brings the same quality and care to people everywhere.

What We Do

Ambessa Coffee is a new company with a simple mission: to offer high-quality Ethiopian coffee at a fair price. We source our beans directly from trusted farmers in Ethiopia, focusing on rich flavor, sustainability, and tradition.

Nedan ser vi utseendet av footern som sedan ska länka till kaffeföretagets riktiga sociala medier, vi kan även se nyhetsregistreringen. Om användaren redan är registrerad dyker ett felmeddelande upp annars dirigeras användaren vidare till en thank-you sida, se nedan.

The footer section features a decorative background with faint coffee beans and leaves. It includes four main categories: SHOP, ABOUT, HELP, and STAY UPDATED. Under SHOP, there are links for Shipping, Returns, and Wholesale. Under ABOUT, there are links for Our story, Privacy policy, and Terms and conditions. Under HELP, there are links for FAQ, Contact us, and Tutorials. The STAY UPDATED section contains a form for entering an email address and a brown "Subscribe" button. Below the form are social media icons for Facebook, Instagram, and TikTok.

SHOP

Shipping
Returns
Wholesale

ABOUT

Our story
Privacy policy
Terms and conditions

HELP

FAQ
Contact us
Tutorials

STAY UPDATED

Email address

Subscribe

© 2025 Ambessa Coffee. All rights reserved

The page has a decorative background with faint coffee beans and leaves. At the top, there are three icons: a menu icon (three horizontal lines), the Ambessa Coffee logo (the word "AMBESSA" above "COFFEE"), and a user profile icon. A central white box contains the text "Welcome to the team" and a message: "You are now subscribing on our newsletters and wont miss out on discounts and new releases. Thank you!"

≡

AMBESSA
COFFEE

Welcome to the team

You are now subscribing on our newsletters and wont miss out on discounts and new releases. Thank you!

Produktsidan från menyn tar oss till denna sida som visar de aktuella kaffesorterna men möjlighet att lägg in i varukorgen.



COFFEE SELECTION



Single Origin

149.00 SEK

Add to cart



House Blend

189.00 SEK

Add to cart



Light Roast

179.00 SEK

Add to cart

SHOP

[Shipping](#)

[Returns](#)

[Wholesale](#)

ABOUT

[Our story](#)

[Privacy policy](#)

[Terms and conditions](#)

HELP

[FAQ](#)

[Contact us](#)

[Tutorials](#)

STAY UPDATED

[Subscribe](#)





COFFEE SELECTION



Single Origin
149.00 SEK
[Add to cart](#)



House Blend
189.00 SEK
[Add to cart](#)

Light Roast
179.00 SEK
[Add to cart](#)

Shopping Bag

Single Origin
Antal: 2
Pris: 149.00 kr

House Blend
Antal: 1
Pris: 189.00 kr

Total: 487 kr

Går vi in på en produkt visas product-detail.html mallen där även en description från SQL databasen bifogas baserad på produktens ID. På den sidan bifogade jag även en tillbaka-knapp för lättare navigering tillbaka.

The image shows a mobile device displaying a product page from the Ambessa Coffee website. At the top, the brand name "AMBESSA" is displayed above "COFFEE" in a stylized font. To the right of the brand name are icons for a lock and a user profile. Below the header, there is a large image of a white coffee bag labeled "House Blend". A back arrow icon is located in the top left corner of the main content area. The title "House Blend" is centered below the image. A detailed description follows: "Our House Blend combines carefully selected Ethiopian beans to create a balanced and smooth coffee. Expect a harmonious mix of fruity and floral aromas with subtle hints of chocolate and spice. This blend is crafted to be versatile and enjoyable any time of day." The price "Price: 189.00 SEK" is listed below the description. A yellow "Add to cart" button is centered at the bottom of the main content area.

AMBESSA
COFFEE

←

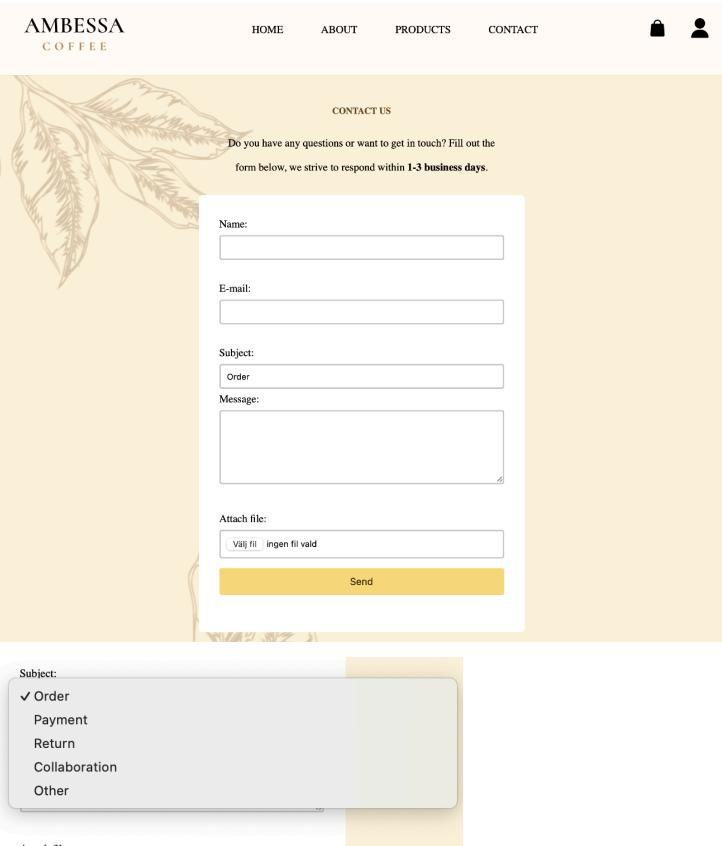
House Blend

Our House Blend combines carefully selected Ethiopian beans to create a balanced and smooth coffee. Expect a harmonious mix of fruity and floral aromas with subtle hints of chocolate and spice. This blend is crafted to be versatile and enjoyable any time of day.

Price: 189.00 SEK

Add to cart

Contact-knappen i menyn tar användaren till detta formulär där valet att bifoga en fil även finns. I subject går det att välja mellan olika alternativ för lättare hantering av mail. Efter lyckat skickat mail får användaren en alert i form av “email-sent” och formuläret rensas.



AMBESSA
COFFEE

HOME ABOUT PRODUCTS CONTACT  

CONTACT US

Do you have any questions or want to get in touch? Fill out the form below, we strive to respond within **1-3 business days**.

Name:

E-mail:

Subject:

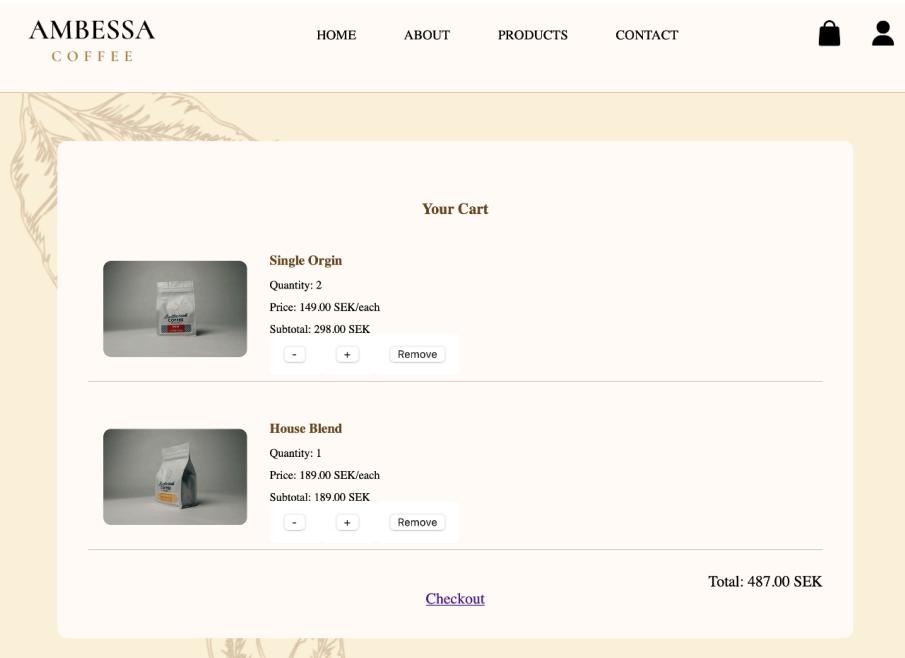
Message:

Attach file:

Send

Subject:
 Order
 Payment
 Return
 Collaboration
 Other

Nedan kan vi se hur det ser ut när varor finns i varukorgen och sedan checkout + bekräftelse. När en beställning är lagd rensas kundvagnen och det står "Your cart is empty.".



AMBESSA
COFFEE

HOME ABOUT PRODUCTS CONTACT  

Your Cart

Single Origin

 Quantity: 2
 Price: 149.00 SEK/each
 Subtotal: 298.00 SEK

House Blend

 Quantity: 1
 Price: 189.00 SEK/each
 Subtotal: 189.00 SEK

Total: 487.00 SEK

[Checkout](#)

Checkout

[←](#)

Full Name

Email

Phone Number

Shipping Address

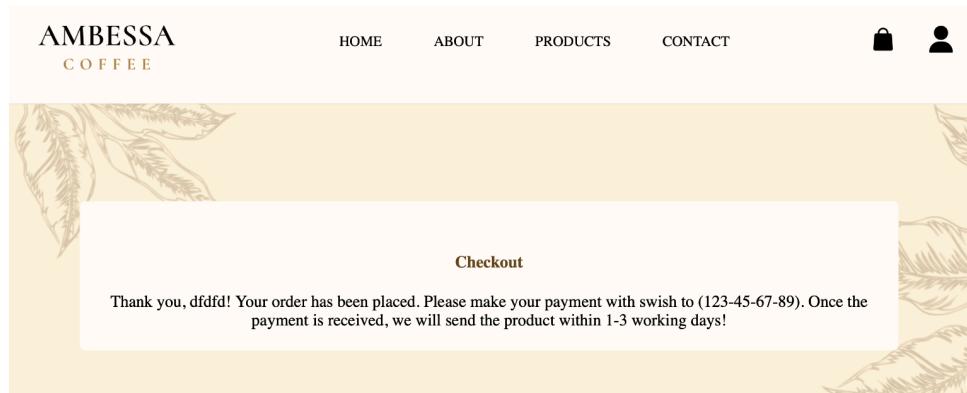
Postal Code

City

City

Payment Method

Place Order



Login och register-sidan ser ut som på bilderna nedan. Vid fel lösenord samt icke-registrerad email dyker felmeddelande upp. Vid lyckad inloggning, det vill säga att uppgifterna stämmer överens med lagrad data i SQL databasen och tabellen "users" kommer användaren direkt till ens profil-dashboard. Här kan användaren se eventuella ordar om lagrade finns i orders kopplat till användarens email, (se funktioner i bild 9). Användaren har tillgång till dessa uppgifter så länge hen är inloggad, på profilsidan kan användaren även logga ut.

The image contains two screenshots of the Ambessa Coffee website, both featuring a light beige background with faint coffee bean illustrations.

Login Page:

- Header:** AMBESSA COFFEE, navigation links: HOME, ABOUT, PRODUCTS, CONTACT, a lock icon, and a user profile icon.
- Form:** A white rectangular form with fields for Email (with placeholder 'Email' and a dropdown arrow) and Password (a blank input field).
- Buttons:** A yellow 'Login' button at the bottom of the form.
- Text:** 'Not registered yet? [Create an account here.](#)'

Register Page:

- Header:** AMBESSA COFFEE, navigation links: HOME, ABOUT, PRODUCTS, CONTACT, a lock icon, and a user profile icon.
- Form:** A white rectangular form with fields for First Name (a blank input field), Last Name (a blank input field), Email (a blank input field), Password (a blank input field), and Confirm Password (a blank input field).
- Buttons:** A yellow 'Register' button at the bottom of the form.
- Text:** 'Already have an account? [Login here.](#)'

Profile Dashboard

Welcome, gina anyim!

- **Email:** ginaanyim@hotmail.com

- **Order History:**

- Date: 2025-08-14 04:05:25
Single Origin (x2)
Light Roast (x1)

Date: 2025-08-14 00:16:22

House Blend (x1)

[Logout](#)