

# E-handel - Gesällprov

Anna Strömberg, anst5816

Institutionen för data-  
och systemvetenskap

*Vetenskaplig metodik och kommunikation inom data- och  
systemvetenskap*

Gesällprov rapport

Webbutveckling II

Vårterminen 2025



# Innehållsförteckning

<b>Instruktion</b>	<b>3</b>
<b>Sammanfattning</b>	<b>3</b>
<b>Framtagande</b>	<b>3</b>
Idé och mål	3
Framtagande av wireframes i Figma	3
Framtagande av fullständig design i Figma	10
Kodning av grundstruktur och stil med HTML5 och CSS3	17
Kodning av interaktivitet och skapandet av databas med PHP och MySQL	17
Förändring från skiss till färdig produkt	17
<b>Form</b>	<b>19</b>
Grundläggande struktur, stil och layout (HTML5, CSS3)	19
Server-side funktionalitet (PHP)	22
Användande av JavaScript	25
Hantering av lagring (phpMyAdmin)	26
Helheten	28
<b>Funktion</b>	<b>29</b>
Användarens första interaktion	29
Navigering och interaktioner	30
Registrering och inloggning	30
Produktvisning och kundvagn	31
Checkout och beställningshantering	31

## Instruktion

**OBS! Börja genom att klicka på "Startpage.html"** för att hemsidan ska se ut som jag vill, märkte att CSS inte ville ladda ifall man inte började på en html-fil.

## Sammanfattning

Webbsidan jag utvecklat är en e-handelsplattform där kunder kan köpa cupcakes och anpassa sina beställningar med personliga meddelanden. Den är designad för att ge en enkel och användarvänlig upplevelse, där kunder kan skapa ett konto, logga in och välja mellan sex olika cupcake-smaker. I kundvagnen har kunder möjlighet att skriva ett kort meddelande, som sedan kommer att spridas på cupcakesen. Webbplatsen består av en startsida, en produktvisningssida och en kundvagn, med stöd för registrering och inloggning. Designen är stilren och responsiv tack vare HTML5 och CSS3, vilket innebär att den fungerar på alla enheter. Med PHP skapas en interaktiv shoppingupplevelse, där kunder enkelt kan registrera sig, logga in och göra sina köp. Systemet använder MySQL för att lagra kunddata, produktinformation och ordrar, vilket gör att webbplatsen fungerar effektivt. Den fokuserar på att ge kunderna möjlighet att skapa en personlig touch på sina beställningar samtidigt som de njuter av en enkel navigering genom sidan.

## Framtagande

Utvecklingen av min e-handelsplattform genomgick flera steg, bland annat använde jag både designverktyget Figma, som är optimalt för planering av design och Visual Studio Code som utvecklingsmiljö. Detta för att skapa en funktionell och estetiskt tilltalande webbsida.

### Idé och mål

Vad jag ville utveckla hade jag inte riktigt kommit på när jag väl var framme vid att påbörja mitt gesällprov. Läste igenom de exempel på vad man kan utveckla för att se ifall någon idé poppade upp. När jag läste "Komplett SQL-genererad e-affär" så visste jag att det kan vara kul att göra. Syftet blev att skapa en lättnavigerad och visuellt tilltalande e-handelsplattform. Från början visste jag inte vad för slags produkter sidan skulle sälja, men medan jag skissade på wireframes i Figma så poppade idén kring personaliserade bakverk upp. Detta kändes som något roligt och jag kunde vara lite mer kreativ i min stil och färgval för sidan, det vill säga göra den lite mer "roligare" och mer "galen". Så jag beslutade mig för att göra en e-handelsplattform för personaliserade cupcakes.

### Framtagande av wireframes i Figma

Framtagandet av wireframes i Figma pågick innan min slutidé var satt i sten. Så de första sidorna som Startpage, Register, Login, Cart och Pay var de första jag tog fram då dessa är

ganska homogena sidor som ofta ser lika ut på varje e-handelssida. När jag väl fick idén om personaliserade cupcakes så kunde jag utforma wireframesidan för produkter och kunde visualisera hur jag ville visa produkterna, cupcakes. Alla wireframes var enkla svartvita skisser som visar på ungefär hur strukturen/designen kommer se ut.

**Startpage.html Wireframe:**

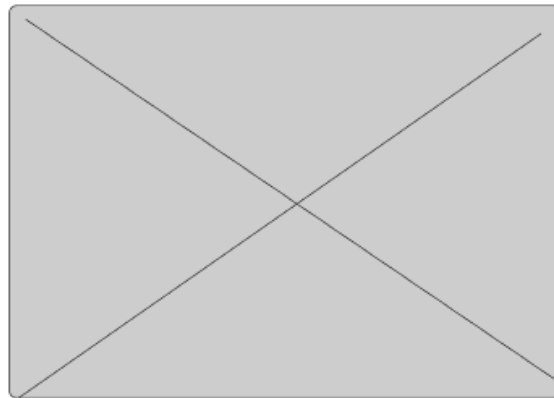
Home

Login

Products

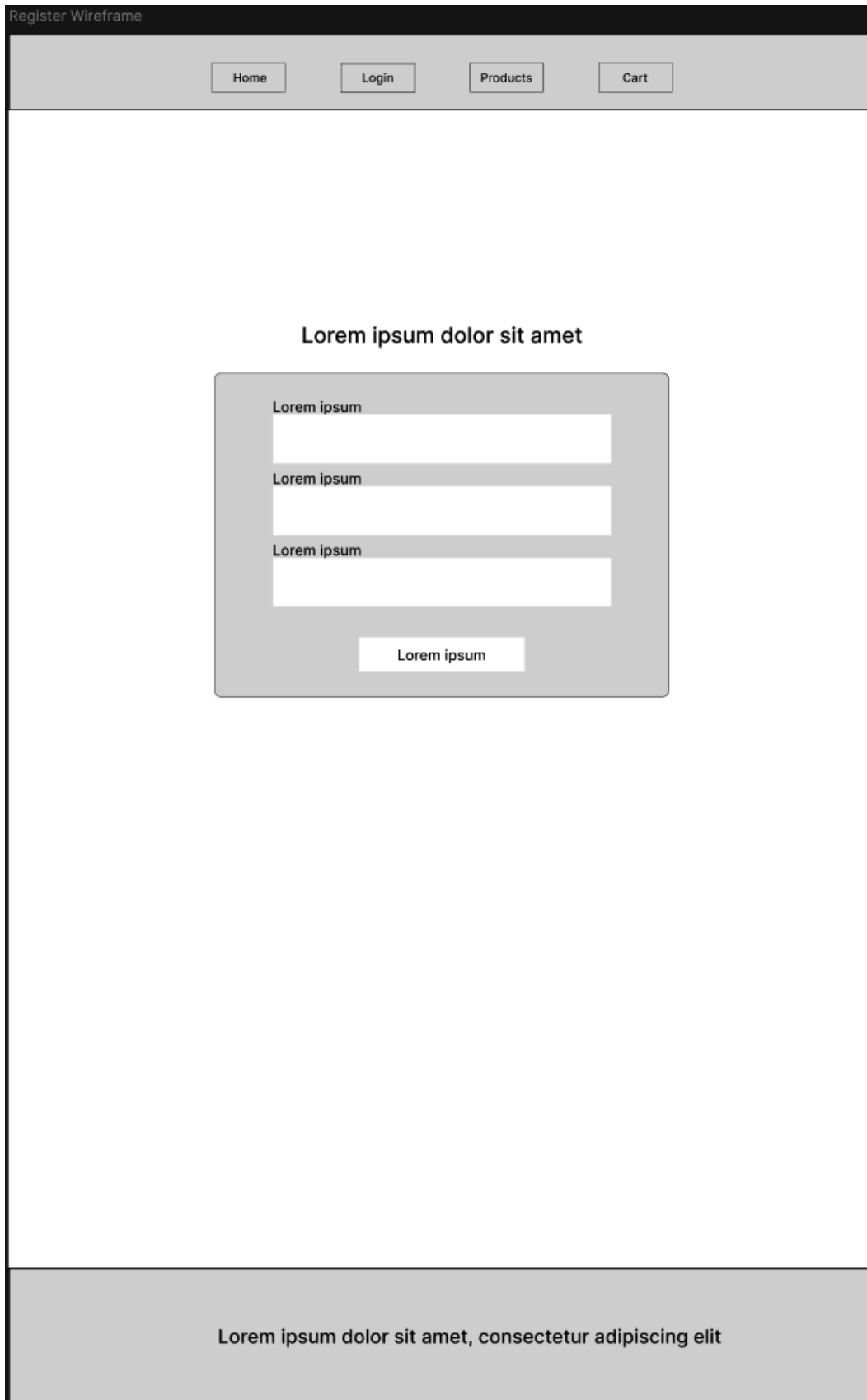
Cart

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit

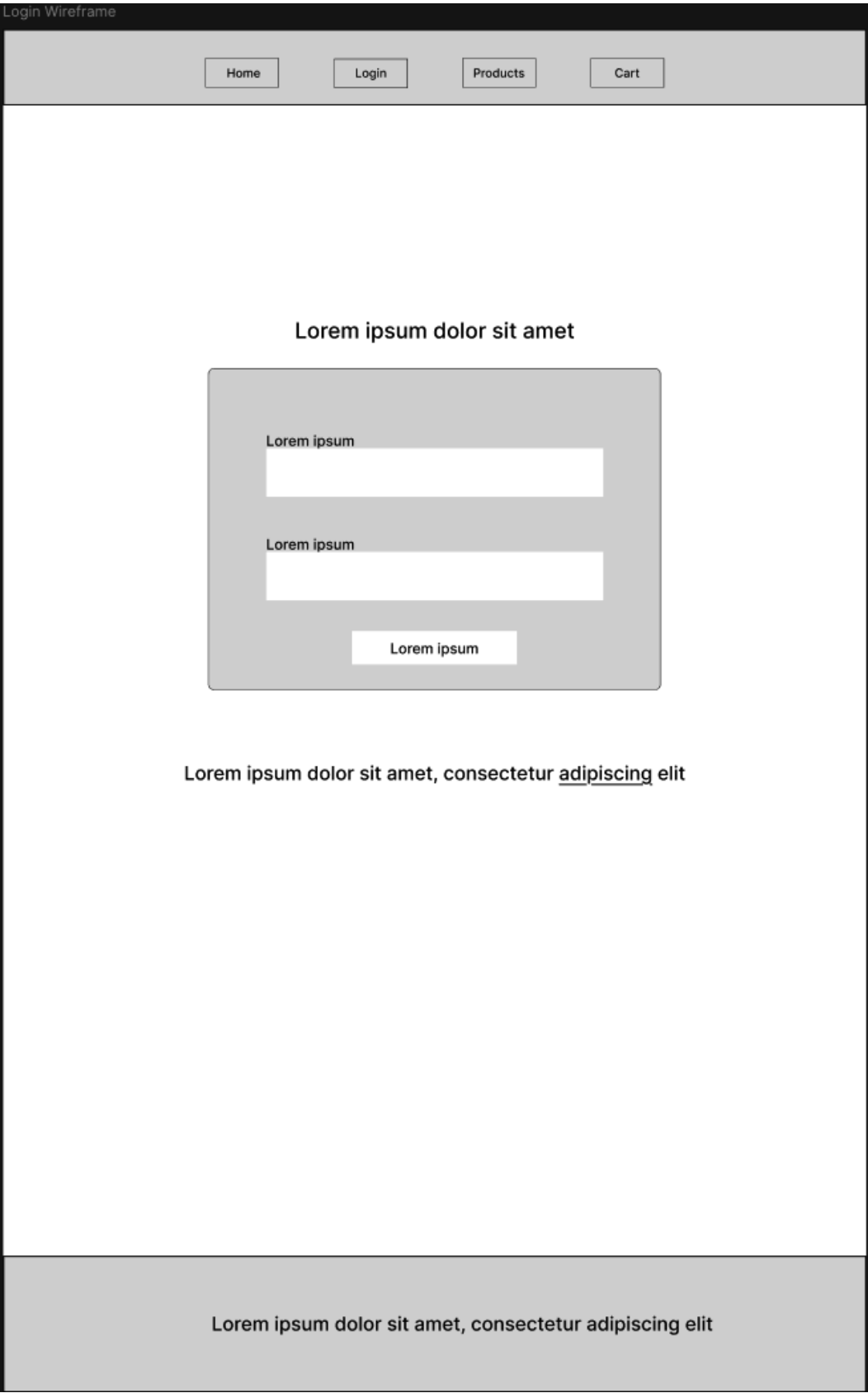


Lorem ipsum dolor sit amet, consectetur adipiscing elit

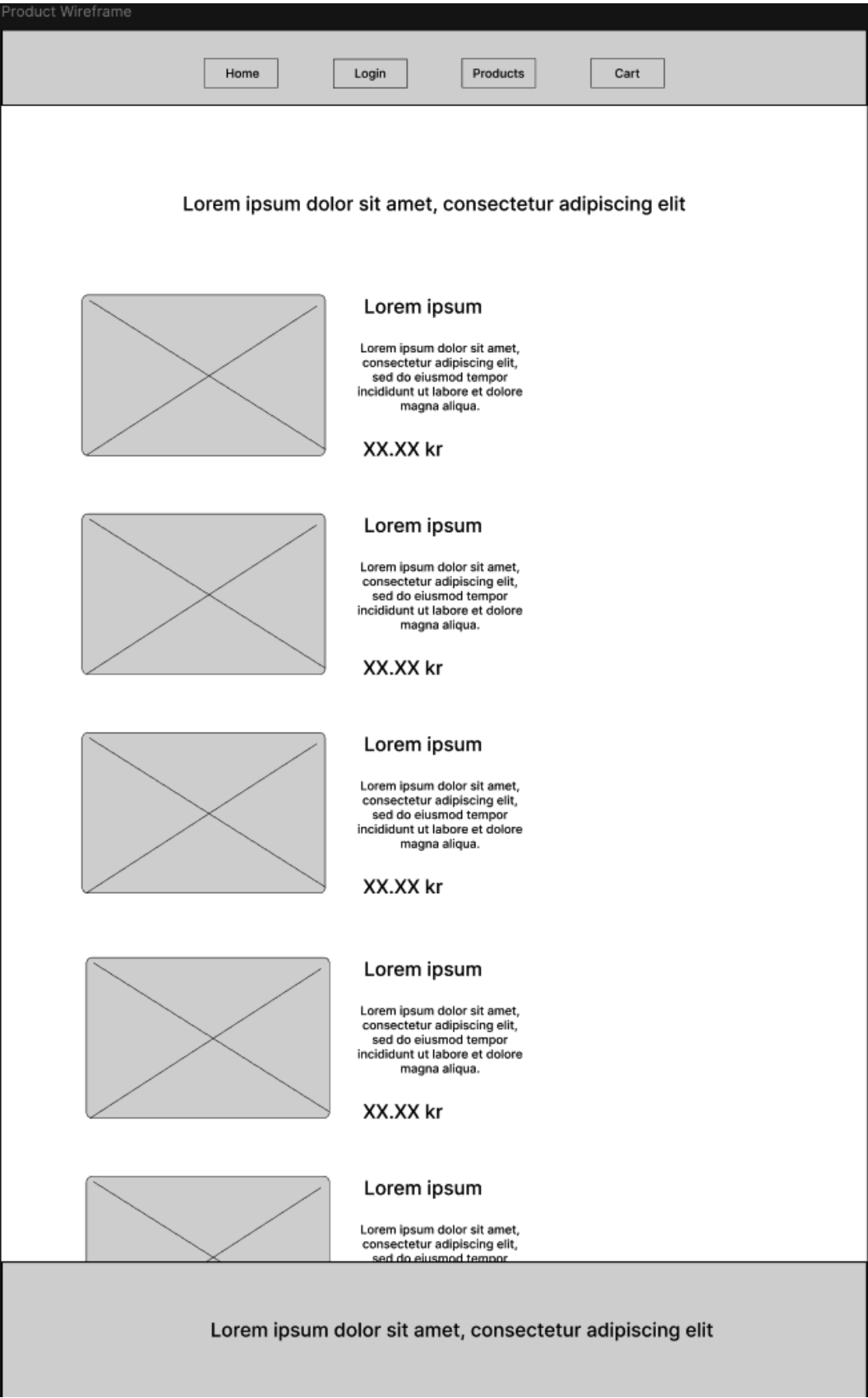
**Register.html Wireframe:**



Login.html Wireframe:



Product.html Wireframe:



Cart.html Wireframe:





Pay.html Wireframe:

Home

Login

Products

Cart

Lorem ipsum

Lorem ipsum	XX	XX.XX kr
Lorem ipsum	XX	XX.XX kr
Lorem ipsum	XX	XX.XX kr
Lorem ipsum	XX	XX.XX kr

Lorem ipsum      XX.XX kr

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit

### Framtagande av fullständig design i Figma

När jag blev nöjd med hur mina wireframes såg ut och själva skissen på hur strukturen skulle se ut så gick jag vidare och skapade den mer detaljerade designen i Figma. Det är i denna del färger, typsnitt, bilder och andra designelement bestämdes. Jag visste att jag ville använda en rosa bakgrundsfärg för att göra webbsidan lite "gullig" då det är en e-handelssida för cupcakes. Utgick från en ganska basic design för login- och registerformulär då fokuset låg mer på produktvisningen. Jag bestämde även bilder för varje produkt och vad för typer av cupcakes jag ville ha, samt en kort sammanfattning som ska få kunden att få en uppfattning kring smak.

## Fullständig design Startpage.html:



## Fullständig design Register.html:

Register Final

Home Login Products Cart

### Register an account

Username

E-mail

Password

Register

© 2025 Anna E-Commerce LLC

## Fullständig design Login.html:

Login Final

Home Login Products Cart

# Login

E-mail

Password

Login

Don't have an account? [Register here!](#)


© 2025 Anna E-Commerce LLC

## Fullständig design Product.html:

Product Final

HomeLoginProductsCart

Check out our delicious cupcakes below:




**Red Velvet Cupcake**

A delicious red velvet cupcake with cream cheese frosting

Price: 30.99 kr

Quantity:  ↑ ↓ Add to Cart




**Vanilla Cupcake**

A light and fluffy vanilla cupcake topped with rich vanilla buttercream frosting

Price: 29.99 kr

Quantity:  ↑ ↓ Add to Cart




**Coffee Cupcake**

A delicious coffee-flavored cupcake with smooth coffee frosting, perfect for coffee lovers

Price: 34.99 kr

Quantity:  ↑ ↓ Add to Cart



**ChocoMint Cupcake**

A moist chocolate cupcake topped with refreshing mint frosting and a hint of chocolate chips

Price: 32.99 kr

Quantity:  ↑ ↓ Add to Cart

© 2025 Anna E-Commerce LLC

## Fullständig design Cart.html:

Cart Final

HomeLoginProductsCart

### Your Cart

Red Velvet Cupcake	1st	30.99 kr
Red Velvet Cupcake	1st	30.99 kr
Vanilla Cupcake	1st	29.99 kr
Vanilla Cupcake	1st	29.99 kr

**Total:** 121.96 kr

Your Cupcake Message:

Happy Birthday...

Go To Checkout

© 2025 Anna E-Commerce LLC



## Fullständig design Pay.html:

Pay Final

HomeLoginProductsCart

### Order Summary

Red Velvet Cupcake	1st	30.99 kr
Red Velvet Cupcake	1st	30.99 kr
Vanilla Cupcake	1st	29.99 kr
Vanilla Cupcake	1st	29.99 kr

Your Message: "Happy Birthday"

**To Pay: 121.96 kr**

Confirm and Pay

© 2025 Anna E-Commerce LLC

### Kodning av grundstruktur och stil med HTML5 och CSS3

Efter att den fullständiga designen var klar och jag kände mig nöjd började jag koda webbsidan. Jag använde mig av HTML5 för att skapa de olika strukturella delarna av sidan. När den grundläggande strukturen var på plats lade jag till CSS-styling för att förbättra sidans visuella design och för att få den att efterlikna den fullständiga designen jag skapat i Figma.

### Kodning av interaktivitet och skapandet av databas med PHP och MySQL

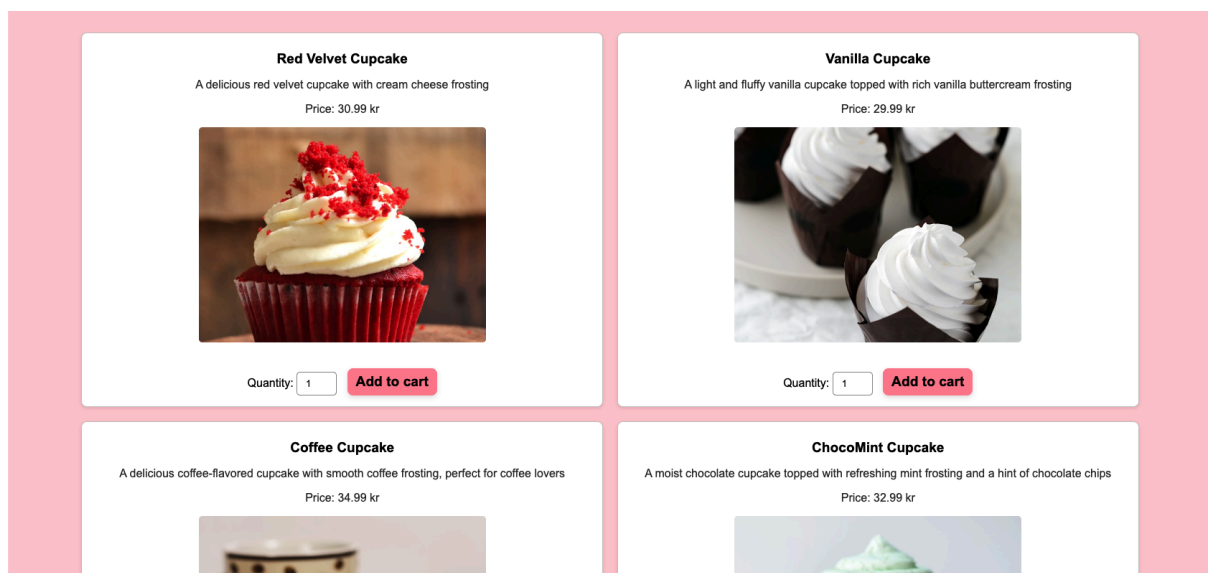
Webbplatsen använder PHP och MySQL för att hantera användardata, produkter och beställningar. MySQL-databasen innehåller tre huvudtabeller: products, user och orders, som lagrar relevant information om produkter, användare och deras beställningar. PHP-skript hanterar användarregistrering, inloggning/utloggning, sammanställer kundvagn och visar produkter dynamiskt på sidan genom att hämta information från databasen, inklusive bilder och produktbeskrivningar. Kundvagnsfunktionen gör det möjligt för användare att lägga till produkter, skriva sitt personliga meddelande och gå vidare till checkout, där beställningar sparas i databasen.

### Förändring från skiss till färdig produkt

Som alltid så förändras produkten under utveckling, det är sällan den färdiga produkten ser ut som den skissade designen i Figma. Jag la till vissa funktioner och jag ändrade hur produkterna visades, samt vissa färgval. Den största design förändringen, som man märker, är nog bakgrundsfärgen, som sätter själva känslan av hemsidan. Jag valde en mer rosa/baby rosa än rosa/lila. Andra små ändringar inkluderar instruktionerna på Startpage.html, en mörkare grå ton för navigation, footer osv.

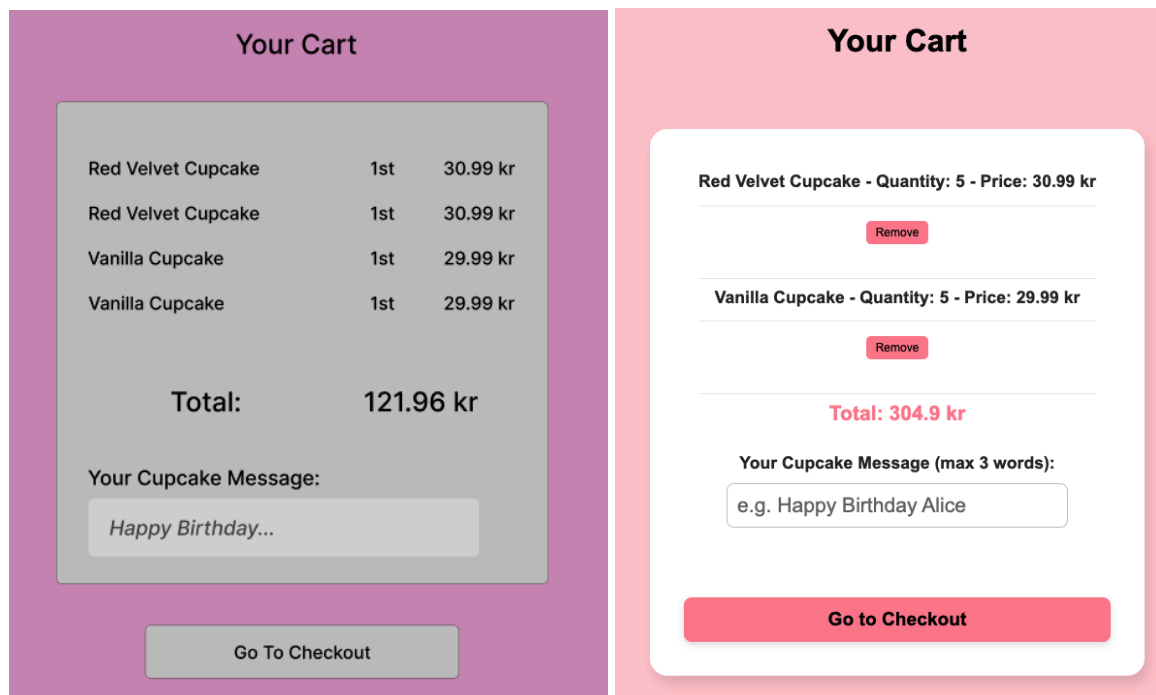
Produktsidan är omstrukturerad och inte i en lång lista, som det var skissat i designen, utan istället med två produkter bredvid varandra per rad. Detta är lite mer användarvänligt och ger en bättre helhetsbild för användaren.

**Första bilden är designskiss och andra bilden är den färdiga hemsidan:**



Ett annat exempel på en förändring jag genomförde var “Remove” knappar i kundvagnen, vilket kändes logiskt att ha med men som jag missat att tänka på under designfasen.

**Till vänster är designskiss och till höger är den färdiga hemsidan:**



Andra förändringar från skiss till färdig produkt var bara små kosmetiska, som exempelvis hur stora knapparna ska vara, hur rundade kanterna är och olika placeringar av element.

## Form

Grundläggande struktur, stil och layout (HTML5, CSS3)

Webbplatsens grundstruktur är byggd med HTML5, vilket säkerställer en välorganiserad struktur. Huvudkomponenter som navigation, produktlistning och användarinformation är definierade med HTML5-element. Dessa HTML-filer används senare av PHP-skript som mallar. En annan central del är CSS3 styling av elementens utseende och interaktivitet. Till exempel används hover-effekter för att förbättra användarens interaktion med knappar och länkar. Färger, typsnitt och spacing är noggrant designade för att skapa en modern och stilren estetik.

## Exempel på grundläggande HTML5-struktur (Startpage.html):

```
WebDevelopment2 > Gesällprov > <> startpage.html > ...
1  <!doctype html>
2  <html lang="se">
3    <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>Anna E-Commerce LLC!</title>
7      <link rel="stylesheet" href="style.css?rnd=12345">
8    </head>
9    <body>
10     <div class="page">
11       <header>
12         <nav>
13           <ul>
14             <li><a href="login.php">Login</a></li>
15             <li><a href="register.php">Register</a></li>
16             <li><a href="product.php">Products</a></li>
17             <li><a href="cart.php">Cart</a></li>
18             <!--logout-->
19           </ul>
20         </nav>
21       </header>
22
23       <h1><!--welcomeuser--></h1>
24       <!--message-->
25       <h3>Here you can buy the best customized cupcakes for a good price!</h3>
26
27       <ol class="instruction">
28         <strong>Instructions</strong>
29         <li>Create an account or log in to get started.</li>
30         <li>Head over to "Products" to browse and choose your favorite cupcakes.</li>
31         <li>Add a personal touch with a custom message during checkout!</li>
32       </ol>
33       
34     </div>
35
36     <footer>
37       <p>&copy; 2025 Anna E-Commerce LLC</p>
38     </footer>
39   </body>
40 </html>
```

### Exempel på CSS3 (style.css):

```
WebDevelopment2 > Gesällprov > # style.css > ...
1  /* Grundläggande styling för sidan */
2  body {
3      font-family: Arial, sans-serif;
4      margin: 0;
5      padding: 0;
6      background-color: ■ rgb(255, 204, 211);
7      display: flex;
8      flex-direction: column;
9      justify-content: flex-start;
10     min-height: 100vh;
11 }
12
13 .page {
14     flex: 1;
15     display: flex;
16     flex-direction: column;
17     align-items: center;
18     text-align: center;
19     padding-top: 60px;
20 }
21
22 section {
23     opacity: 0;
24     transform: translateY(50px);
25     transition: opacity 1.5s ease, transform 1.5s ease;
26 }
27
28 section.in-view {
29     opacity: 1;
30     transform: translateY(0px);
31 }
```

```

158  /* Styling för produkterna */
159  .container {
160      display: grid;
161      grid-template-columns: repeat(2, 1fr);
162      gap: 20px;
163      margin: 20px;
164  }
165
166  .product {
167      border: 1px solid #ccc;
168      border-radius: 8px;
169      padding: 15px;
170      background-color: #fff;
171      box-shadow: 0 2px 5px rgba(0,0,0,0.1);
172      text-align: center;
173  }
174
175  .product img {
176      width: 100%;
177      max-width: 400px;
178      height: auto;
179      max-height: 300px;
180      object-fit: cover;
181      border-radius: 5px;
182      display: block;
183      margin: 0 auto;
184  }
185
186  .product h3 {
187      font-size: 1.2em;
188      margin: 10px 0;
189  }
190
191  .product p {
192      font-size: 1em;
193      color: #333;
194  }

```

### Server-side funktionalitet (PHP)

PHP används för att hantera webbplatsens dynamiska funktionalitet, inklusive användarregistrering, inloggning/utloggning, kundvagn/betala och produktvisning. Användardata lagras i databasen och PHP-skript används för att hämta denna data, bearbeta användarinmatning och uppdatera webbplatsen utan att behöva ladda om sidan.

För produktvisningen används PHP-skript för att hämta produktinformation dynamiskt från databasen. PHP kopplar till MySQL-databasen och hämtar detaljer om varje produkt, såsom namn, beskrivning och pris. Informationen används sedan för att skapa en dynamisk HTML-struktur som visas för användaren. På så sätt kan produkterna uppdateras enkelt i

databasen och ändringarna reflekteras direkt på webbplatsen utan att behöva ändra på själva koden eller ladda om sidan. Denna del av sidan använder även JavaScript för att ladda in sektionen snyggare med en scroll-effekt när de kommer i fokus.

En annan central funktion är användarens registrering och inloggning. Vid registrering sparas användarens uppgifter i databasen, medan vid inloggning autentiseras användarens e-post och lösenord mot de lagrade värdena i databasen. Dessutom används PHP för att hantera kundvagnsfunktionen, där varje produkt som läggs till sparas i PHP-sessionen för att hålla reda på användarens val tills de går till kassan. Väl i kassan där användaren kan skriva in sitt personliga meddelande så sparas detta meddelande i användarens order, som i sin tur sparas i databasens tabell för ordrar.

#### Exempel på centralt PHP-kodavsnitt, användarautentisering (login.php):

```
35 //Hantera formulärdata
36 if ($_SERVER["REQUEST_METHOD"] == "POST") {
37     $email = filter_var(value: trim(string: $_POST["email"]), filter: FILTER_SANITIZE_EMAIL);
38     $password = $_POST["password"];
39     //Hämta kundinformation från databasen baserat på e-post
40     $sql = "SELECT id, username, password FROM user WHERE email = ?";
41     $stmt = $conn->prepare(query: $sql);
42     $stmt->bind_param(types: "s", var: &$email);
43     $stmt->execute();
44     $stmt->store_result();
45     //Kontrollera om kund hittades
46     if ($stmt->num_rows > 0) {
47         $stmt->bind_result(var: &$user_id, vars: &$username, $hashed_password);
48         $stmt->fetch();
49         //Verifiera lösenord
50         if (password_verify(password: $password, hash: $hashed_password)) {
51             $_SESSION["user_id"] = $user_id;
52             $_SESSION["username"] = $username;
53             header(header: "Location: startpage.php");
54             exit();
55         } else {
56             $message = <<<EOD
57             <p>Wrong password!</p>
58             EOD;
59         }
60     } else {
61         $message = <<<EOD
62         <p>User not found!</p>
63         EOD;
64     }
65     $stmt->close();
66 }
```

#### Pseudokod:

Om request-metoden är "POST":

Ta emot och sanera användarens e-postadress

Ta emot användarens lösenord

Skapa SQL-fråga för att hämta användarens id, användarnamn och



```

lösenord baserat på e-post
  Förbered SQL-frågan och binda parametrarna
  Utför SQL-frågan
  Spara resultatet för vidare kontroll

  Om en användare hittas (num_rows > 0):
    Hämta användarens id, användarnamn och hashat lösenord
    från resultatet
    Verifiera det inskickade lösenordet mot det lagrade
    hashade lösenordet
    Om lösenordet är korrekt:
      Sätt användarens id och användarnamn i sessionen
      Omdirigera till startsidan
      Avsluta programmet
    Annars:
      Sätt felmeddelande för felaktigt lösenord
  Annars:
    Sätt felmeddelande för att användaren inte hittades

  Stäng SQL-förberedelsen

```

Exempel på centralt PHP-kodavsnitt, produktinformation (product.php):

```

74 while ($row = $result->fetch_assoc()) {
75     $block = <<<EOD
76
77         <div class="product">
78             <h3>{$row['name']}</h3>
79             <p>{$row['description']}</p>
80             <p>Price: {$row['price']} kr</p>
81         EOD;
82         //Hämta bild om den finns
83         $image_sql = "SELECT image FROM products WHERE id = " . $row['id'];
84         $image_result = $conn->query(query: $image_sql);
85         if ($image_result->num_rows > 0) {
86             $image_row = $image_result->fetch_assoc();
87             //Lägg till bild direkt efter produktinformation
88             $block .= "<p><img src='product.php?id=" . $row['id'] . "' alt='Bild saknas!' /></p>";
89         }

```

Pseudokod:

```

Medan varje rad är lika med resultat från SQL-frågan:
  Skapa en HTML-struktur för produkten:
    Lägg till produktens namn, beskrivning och pris
  Skapa en SQL-fråga för att hämta bilddata för den aktuella
  produkten
  Om bilden finns (num_rows > 0):
    Lägg till produktbilden i HTML-strukturen
    Lägg till den färdiga HTML-strukturen till den

```

## övergripande HTML-sidan

### Slut på loopen

#### Användande av JavaScript

Jag använde även lite JavaScript för att förbättra användarupplevelsen med snygga animationer. IntersectionObserver användes för att skapa scroll-animationen när en sektion (i detta fall produktsidan) blir synlig för användaren. Toast pop-up och AJAX användes för att ge snabb feedback till användaren när de interagerar med webbplatsen (lägga till produkter i kundvagnen), utan att behöva ladda om sidan.

#### Exempel på centralt JavaScript-kodavsnitt (popup.js):

```
12 function showPopup(message) {
13     //Skapa en toast-div
14     const toast = document.createElement("div");
15     toast.classList.add("toast");
16     toast.textContent = message;
17
18     //Lägg till toasten i body
19     document.body.appendChild(toast);
20
21     //Vänta på några sekunder och ta bort toasten
22     setTimeout(() => {
23         toast.classList.add("show"); //Gör toasten synlig
24     }, 100);
25
26     //Ta bort toasten efter 3 sekunder
27     setTimeout(() => {
28         toast.classList.remove("show"); //Döljer toasten
29         document.body.removeChild(toast); //Tar bort toasten från DOM
30     }, 3500);
31 }
```

#### Pseudokod:

Skapa funktion **"showPopup"** med argumentet **"message"**:

- Skapa en ny div-element för toast

- Lägg till CSS-klassen **"toast"** på div-elementet

- Sätt textinnehållet för toasten till det meddelande som skickades in (från add\_cart.php)

- Lägg till toasten i body

Vänta i 100 millisekunder:

- Gör toasten synlig genom att lägga till klassen **"show"** på toasten

Vänta i 3500 millisekunder (3.5 sekunder):  
Dölj toasten genom att ta bort klassen "show"  
Ta bort toasten från body (DOM)

Hantering av lagring (phpMyAdmin)

phpMyAdmin användes för att hantera och administrera MySQL-databasen där all relevant data lagras för webbplatsens funktionalitet.



Databasen E\_Commerce\_db består av tre huvudtabeller som hanterar olika typer av data:

- Tabellen *products*:

Denna tabell lagrar information om de produkter som finns tillgängliga på webbplatsen. Varje produkt har ett unikt id, namn, beskrivning, pris och bild (med filens MIME-typ). Denna tabell gör det möjligt att dynamiskt visa produkter på webbplatsen och uppdatera produktinformation vid behov.

				id	name	description	price	image	mime_type
<input type="checkbox"/>				1	Red Velvet Cupcake	A delicious red velvet cupcake with cream cheese f...	30.99	[BLOB - 86,4 kB]	image/jpeg
<input type="checkbox"/>				2	Vanilla Cupcake	A light and fluffy vanilla cupcake topped with ric...	29.99	[BLOB - 58,0 kB]	image/jpeg
<input type="checkbox"/>				3	Coffee Cupcake	A delicious coffee-flavored cupcake with smooth co...	34.99	[BLOB - 70,9 kB]	image/jpeg
<input type="checkbox"/>				4	ChocoMint Cupcake	A moist chocolate cupcake topped with refreshing m...	32.99	[BLOB - 42,7 kB]	image/jpeg
<input type="checkbox"/>				5	Lemon Cupcake	A zesty lemon cupcake topped with fluffy vanilla f...	29.99	[BLOB - 35,9 kB]	image/jpeg
<input type="checkbox"/>				6	Oreo Cupcake	A rich oreo-chocolate cupcake with creamy vanilla ...	32.99	[BLOB - 58,5 kB]	image/jpeg

- Tabellen *user*:

Användartabellen lagrar information om registrerade användare, inklusive ett unikt id, användarnamn, e-postadress och hashat lösenord. Denna tabell används för autentisering vid inloggning och hantering av användaruppgifter.

**SELECT \* FROM `user`**

☐ Profiling [\[ Redigera infogade \]](#) [\[ Redigera \]](#) [\[ Förklara SQL-kod \]](#) [\[ Skapa PHP-kod \]](#) [\[ Uppdatera \]](#)

☐ Visa alla | Antal rader: 25 | Filtrera rader: Sök i denna tabell | Sort by key: Inget

Extra options

				id	username	email	password
<input type="checkbox"/>				1	admin	admin@gmail.com	\$2y\$10\$vcZC5cpZTo2Aom7T0VPhOuRIhvDMPB2Yt5TDiSS8TVo...
<input type="checkbox"/>				2	root	root@gmail.com	\$2y\$10\$uF7jTzZwEGVA7LIAUvzaE.eWK2W.IRgyVCdVAmWJSaP...
<input type="checkbox"/>				3	katrin	katrin@gmail.com	\$2y\$10\$JoJyavoz7LlhFwUUUqq9JiuEuoFCLqYoL8TbP13im8b...
<input type="checkbox"/>				4	elida	elida@gmail.com	\$2y\$10\$SNz8HDngJugSTaIAOhKI8e2jRBtzSL0ci86o.z1fBnc...
<input type="checkbox"/>				5	theo	theo@gmail.com	\$2y\$10\$wgUHGRbQxRizBcvzJYaN9.O3Wjez.JS9xMiNRGvATR9...

### - Tabellen *orders*:

Ordertabellen hanterar kunders beställningar. Varje order har ett unikt id, koppling till user\_id (den användare som gjort beställningen), product\_id (produkten som beställts), antal (antal av produkten) och ett personligt meddelande som användaren kan skriva till sin order. Tabellen lagrar även tidsstämpeln för när beställningen lades.

**SELECT \* FROM `orders`**

☐ Profiling [\[ Redigera infogade \]](#) [\[ Redigera \]](#) [\[ Förklara SQL-kod \]](#) [\[ Skapa PHP-kod \]](#) [\[ Uppdatera \]](#)

☐ Visa alla | Antal rader: 25 | Filtrera rader: Sök i denna tabell | Sort by key: Inget

Extra options

				id	user_id	product_id	quantity	message	order_date
<input type="checkbox"/>				1	1	1	3	NULL	2025-04-23 13:27:11
<input type="checkbox"/>				2	1	2	1	NULL	2025-04-23 13:27:11
<input type="checkbox"/>				3	1	1	3	NULL	2025-04-23 13:46:45
<input type="checkbox"/>				4	1	4	2	NULL	2025-04-23 13:46:45
<input type="checkbox"/>				5	2	4	6	NULL	2025-04-23 14:12:32
<input type="checkbox"/>				6	3	4	1	NULL	2025-04-23 15:07:14
<input type="checkbox"/>				7	3	3	2	NULL	2025-04-23 15:07:14
<input type="checkbox"/>				8	1	1	1	NULL	2025-04-23 15:10:49
<input type="checkbox"/>				9	2	1	1	NULL	2025-04-29 11:42:10
<input type="checkbox"/>				10	2	2	1	NULL	2025-04-29 11:42:10
<input type="checkbox"/>				11	2	5	3	NULL	2025-04-29 11:56:54
<input type="checkbox"/>				12	2	6	1	NULL	2025-04-29 11:56:54
<input type="checkbox"/>				13	3	5	8	Happy Birthday Martin	2025-04-29 12:25:28
<input type="checkbox"/>				14	4	3	3	Congratz Emmy	2025-04-29 12:27:08
<input type="checkbox"/>				15	4	4	3	Congratz Emmy	2025-04-29 12:27:08
<input type="checkbox"/>				16	4	1	10	Tea Party	2025-04-29 12:58:23
<input type="checkbox"/>				17	4	6	6	Tea Party	2025-04-29 12:58:23
<input type="checkbox"/>				18	4	5	10	Tea Party	2025-04-29 12:58:23
<input type="checkbox"/>				19	4	2	6	Tea Party	2025-04-29 12:58:23
<input type="checkbox"/>				20	1	6	20	LOVE YOU	2025-04-29 13:34:06
<input type="checkbox"/>				21	1	2	10	Baby Shower	2025-05-05 14:56:56
<input type="checkbox"/>				22	5	3	40	Kakor till barnen	2025-05-07 20:52:21
<input type="checkbox"/>				23	1	2	5	Best Sister	2025-05-09 14:31:12

PHP används för att interagera med databasen och skicka SQL-frågor som hämtar och bearbetar data. Vid exempelvis användarregistrering och inloggning används SQL-frågor för att verifiera användaruppgifter genom att hämta relevant information från databasen och jämföra dessa med användarens inmatade data. Vid hantering av kundvagnen lagras produktens id och kvantitet i PHP-sessionen och när kunden går vidare till checkout kopplas dessa produkter till en ny beställning som sparas i orders-tabellen.

Helheten

#### Use Case:

1. Användaren laddar sidan
2. Webbssidan laddar Startpage.html
3. Användaren läser instruktionerna och väljer att registrera sig
4. Efter registrering loggar användaren in
5. Användaren går till Products.php och väljer vilka cupcakes hen vill köpa samt antal
6. Användaren lägger till produkterna i Cart.php
7. Användaren går till Cart.php och skriver in sitt personliga meddelande
8. Användaren väljer "Go To Checkout"
9. Webbssidan laddar in Pay.php
10. Användaren klickar på "Confirm and Pay"
11. Beställningen slutförs
12. Användaren väljer att logga ut

#### Pseudokod:

##### Start:

Ladda webbsidan

Ladda Startpage.html

Om användaren ser instruktioner och väljer att registrera sig:

Ladda Register.php

Om användaren skickar in registreringsformuläret:

Spara användardata i databasen

Visa meddelande om lyckad registrering

Ladda Login.php

Om användaren loggar in:

Ta emot användarens e-post och lösenord

Hämta användarens data från databasen baserat på e-post

Om användarens uppgifter är korrekta:

Skapa en session och spara användarens ID

Ladda Products.php

Annars:  
Visa felmeddelande om användaren inte hittades eller fel lösenord

Om användaren går till `Products.php`:  
Visa lista på cupcakes  
Användaren väljer vilka cupcakes hen vill köpa och anger antal  
Lägg till de valda produkterna i kundvagnen (`Cart.php`)

Om användaren går till `Cart.php`:  
Visa kundvagnens innehåll  
Användaren skriver in sitt personliga meddelande (t.ex. "Happy Birthday Alice")  
Spara meddelandet för produkten

Om användaren väljer "Go To Checkout":  
Ladda `Pay.php`

Om användaren klickar på "Confirm and Pay":  
Skapa en order i databasen med alla produkter och meddelanden  
Visa bekräftelse på slutförd beställning

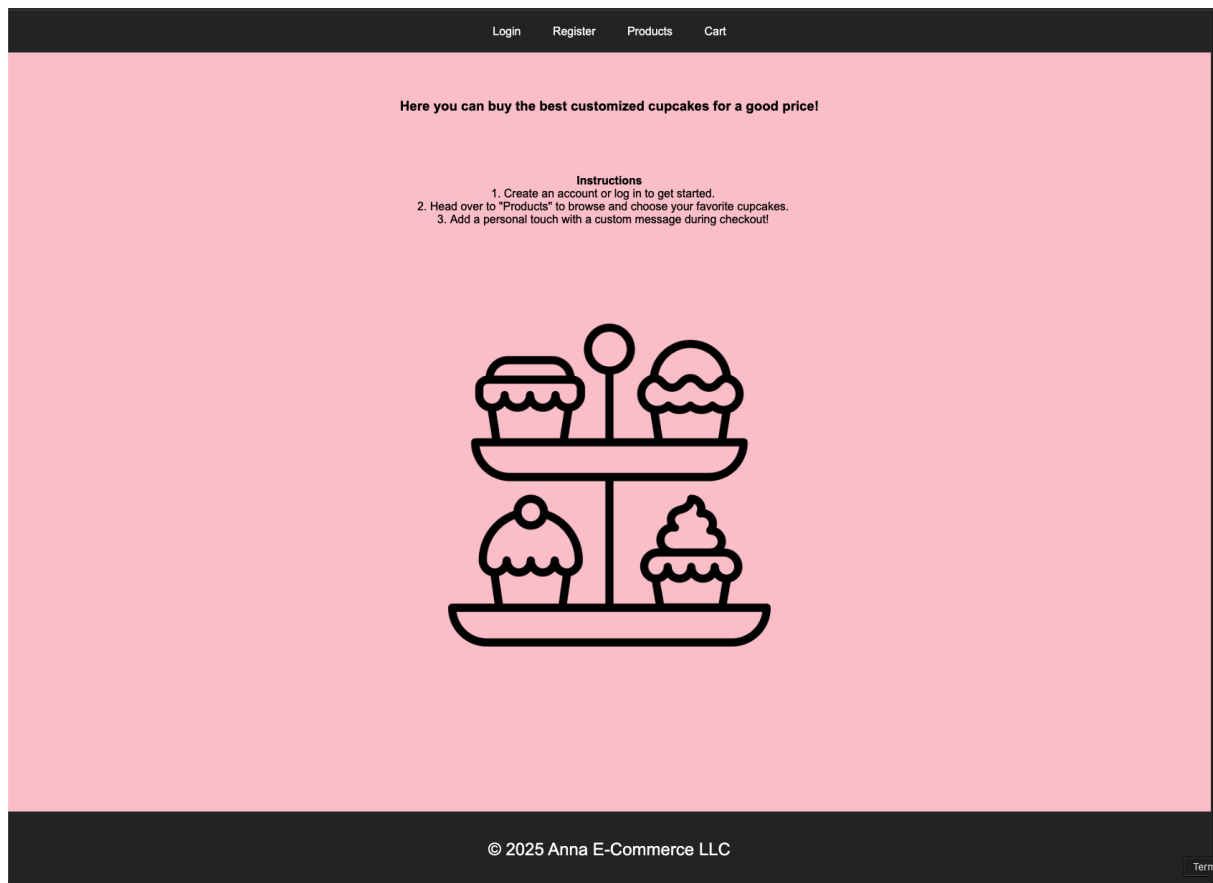
Om användaren loggar ut:  
Rensa sessionen  
Ladda `Startpage.php`  
Visa meddelande om att användaren har loggat ut  
Slut.

## Funktion

### Användarens första interaktion

När användaren först kommer till webbsidan ser de en startsida med en navigationsbar längst upp och instruktioner. Användaren kan välja att logga in, registrera sig, gå till produkterna eller kundvagnen. Startsidan innehåller instruktioner som hjälper användaren att förstå webbplatsens syfte och hur man registrerar sig eller loggar in för att komma åt de interaktiva funktionerna.

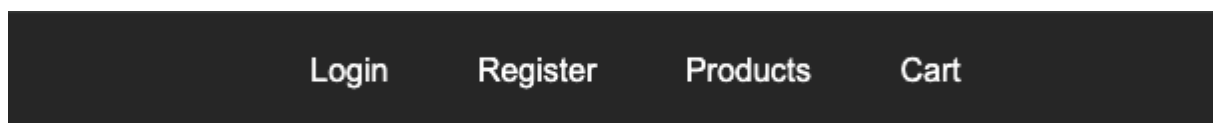
### Startpage.html:



### Navigering och interaktioner

Användaren kan enkelt navigera genom hemsidan via en fast navigeringsmeny som tillåter snabb åtkomst till "Login", "Register", "Products" och "Cart". Däremot om användaren klickar på "Products" eller "Cart" har jag implementerat så att man måste först logga in som användare innan man kan se allt. När användaren väl har antingen registrerat sig eller loggat in så kan man klicka på "Products" och "Cart" och se innehållet.

#### Navigationsbar:



### Registrering och inloggning

Användare kan skapa ett konto genom att ange ett användarnamn, e-post och lösenord. Vid inloggning verifieras e-post och lösenord mot databasen för att ge åtkomst. Detta ger åtkomst till innehållet på hela hemsidan och väl inloggad kan användaren välja att logga ut med knappen "Logout" som bara dyker upp ifall man är inloggad.

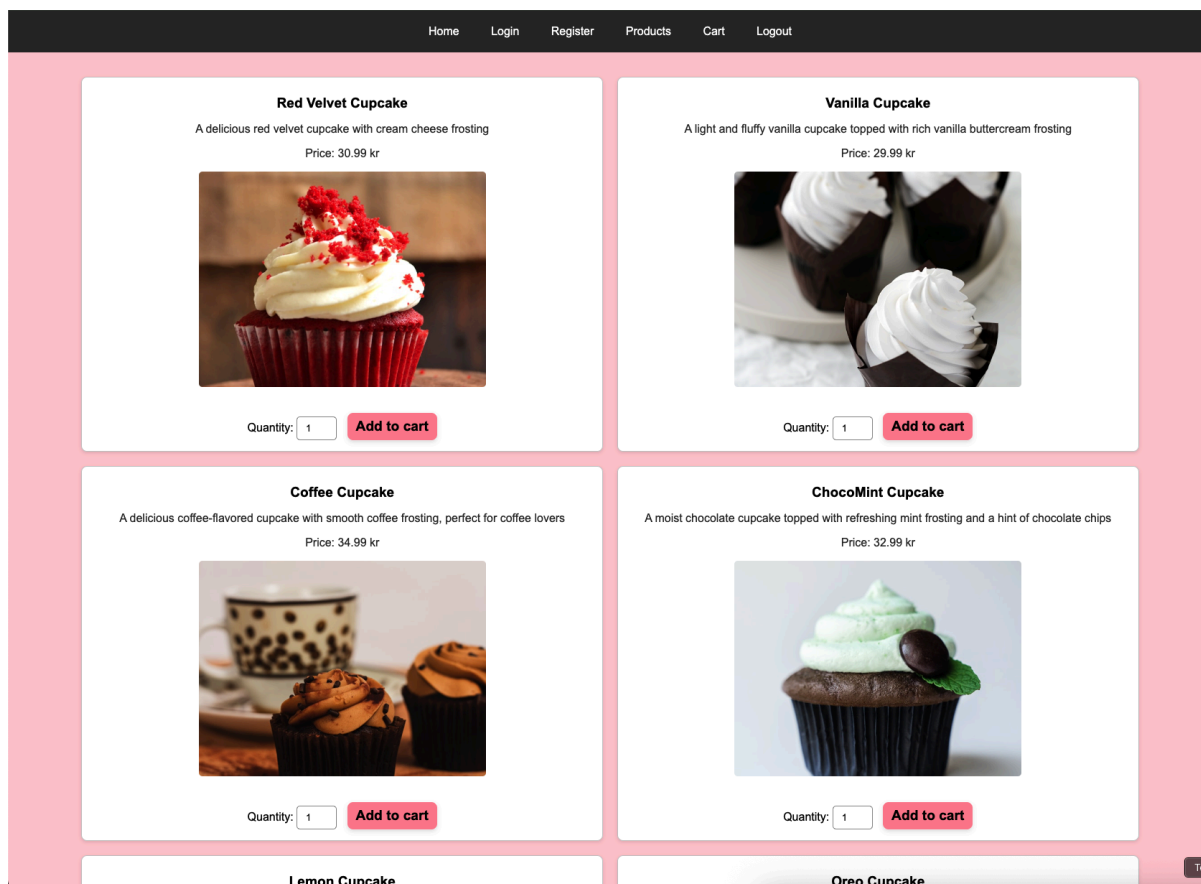
**Första bilden är navigationsbar innan inlogg och andra bilden är navigationsbar efter inlogg:**

[Login](#)[Register](#)[Products](#)[Cart](#)[Login](#)[Register](#)[Products](#)[Cart](#)[Logout](#)

### Produktvisning och kundvagn

När användaren väljer att gå till "Products" genom navigationsbaren så laddas produkterna in dynamiskt med hjälp av JavaScript tillsammans med PHP-skriptet. Detta gör att produkterna laddas med en scroll-effekt när de kommer i fokus. Användaren kan läsa vad för typ av cupcake/smak på cupcake, en beskrivning av cupcaken och pris per antal. Användaren kan välja hur många antal av den specifika cupcaken som den vill lägga i kundkorgen. Produkterna lagras i sessionen tills användaren går vidare till kassan.

### Produktvisning:



### Checkout och beställningshantering

Vid checkout (Cart) sammanställs kundens beställning där de kan se antal, pris och totala summan samt lägga till sitt personliga meddelande som kommer att spridas på deras cupcakes. Efter att användaren har bekräftat och genomfört beställningen lagras



informationen i databasen för att skapa en orderhistorik och användaren får en bekräftelse på sin beställning.

### Your Cart

ChocoMint Cupcake - Quantity: 15 - Price: 32.99 kr

---

Remove

---

**Total: 494.85 kr**

Your Cupcake Message (max 3 words):

Mint Condition

Go to Checkout

### Order Summary

**Thank you for your order!**

**Your order has been placed successfully.**

---

Return to homepage