

# Presentazione del mio e-commerce

*Dolcini*

**CORSO DI WEBDESIGN 2**

**A. A. 2022-2023**

Matricola **834**

Docente **Matteo Baccan**

## Aree di lavoro

Per questo progetto ho creato un sito e-commerce sulla vendita di dolci, composto da **4 pagine collegate tra di loro:**

- Home
- Pagina prodotto
- Carrello
- About us

Queste pagine sono accomunate da una **headline** e da un **footer** ricorrenti in ogni pagina.

# Strumenti usati

Per programmare le varie pagine di questo sito mi sono affidata a **Codespace**, l'ambiente di sviluppo gratuito di **Github** basato sul cloud che consente agli sviluppatori di creare, testare e distribuire le loro applicazioni direttamente dalla piattaforma GitHub.

CodeSpace **semplifica e accelera il processo di sviluppo** delle applicazioni e **promuove la collaborazione tra gli sviluppatori** all'interno della piattaforma GitHub.



# Strumenti usati

Per vedere l'anteprima del mio sito, ho collegato il mio repository a questo link di netlify:

<https://github.com/ottafregu/ecommerce-freguglia-dolcini>

**Netlify** è una società di cloud *computing remote-first* che offre una piattaforma di sviluppo che include build, deployment e servizi di back-end serverless per applicazioni Web e siti Web dinamici. La piattaforma è costruita su standard web aperti, rendendo possibile l'integrazione di strumenti di compilazione, framework web, API e varie tecnologie web in un flusso di lavoro unificato per gli sviluppatori.



## Strumenti usati

Per la creazione di alcune parti del mio sito, ho utilizzato **Bootstrap**, in modo da aiutarmi a gestire in maniera più responsive gli spazi.

Bootstrap è un framework di sviluppo web gratuito e open source, progettato per facilitare il processo di sviluppo web di siti responsive e mobile-first, fornendo una raccolta di sintassi per i modelli di progettazione.



# Elementi comuni

*Headline & Footer*



## Headline

Nella parte sinistra della headline troviamo un **menù a comparsa** che, al click sulle tre lineette, richiama l'evento 'activateMenu', permettendo così alle 3 fasce '.menu-top', '.menu-middle' e '.menu-bottom' di espandersi e rendendo così visibile sia il loro link interno '.menu-text', sia la X '.menu-close' per chiudere l'intero menù e farlo tornare alla forma iniziale.

# Headline

Codice in js del menu

```
/*menu*/
var Menu = {
  el: {
    menu: $('.menu'),
    menuTop: $('.menu-top'),
    menuClose: $('.menu-close'),
    menuMiddle: $('.menu-middle'),
    menuBottom: $('.menu-bottom'),
    menuText: $('.menu-text')
  },
  init: function() {
    Menu.bindUIActions();
  },
  bindUIActions: function() {
    Menu.el.menu
      .on(
        'click',
        function(event) {
          Menu.activateMenu(event);
          event.preventDefault();
        }
      );
  },
  activateMenu: function() {
    Menu.el.menuTop.toggleClass('menu-top-expand expand');
    Menu.el.menuMiddle.toggleClass('menu-middle-expand expand');
    Menu.el.menuBottom.toggleClass('menu-bottom-expand expand');
    Menu.el.menuText.toggleClass('menu-text-expand');
    Menu.el.menuClose.toggleClass('menu-close-visible');
  },
  //Stop menu item click closing the menu
  $(".menu .menu-global").click(function(e) {
    e.stopPropagation();
  });
}
Menu.init();
```



## Headline

Al centro della headline troviamo il **logo del brand** che funge anche da collegamento alla home del sito, lavorando come se fosse un link.



## Headline

Nella parte destra della headline troviamo il **carrello** in cui, grazie alla funzione 'aggiungiAlCarrello' e 'rimuoviDalCarrello', associate ai tasti presenti sotto ai box prodotti, possiamo vedere il numero di questi inseritovi dentro.

Inoltre, questa icona funge da collegamento alla pagina del carrello stesso.

## Footer

Nel footer troviamo la mappa, i contatti e i social: **tre elementi inline** che, allo stringersi della pagina, si posizionano uno sotto l'altro grazie ad una **media query**.



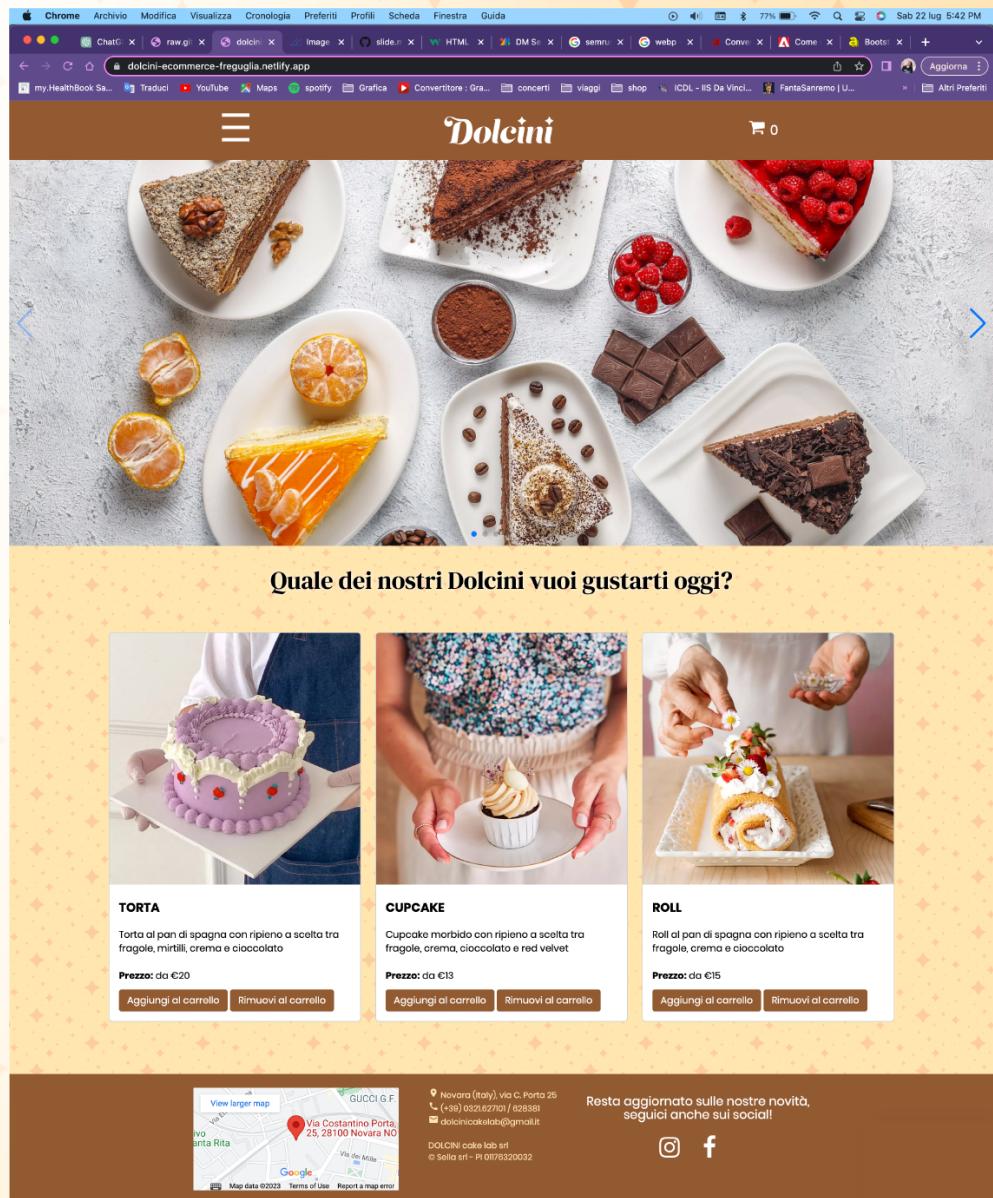
# Pagine singole

*Home - Prodotto - Carrello - About us*

## Home page

Nella home page troviamo uno slider, un titolo e le tre caselle prodotto.

- Lo slider, cui codice nativo è stato preso da <https://swiperjs.com/>, contiene 6 immagini di dolci che scorrono una dopo l'altra grazie alla funzione Javascript 'autoplay'.



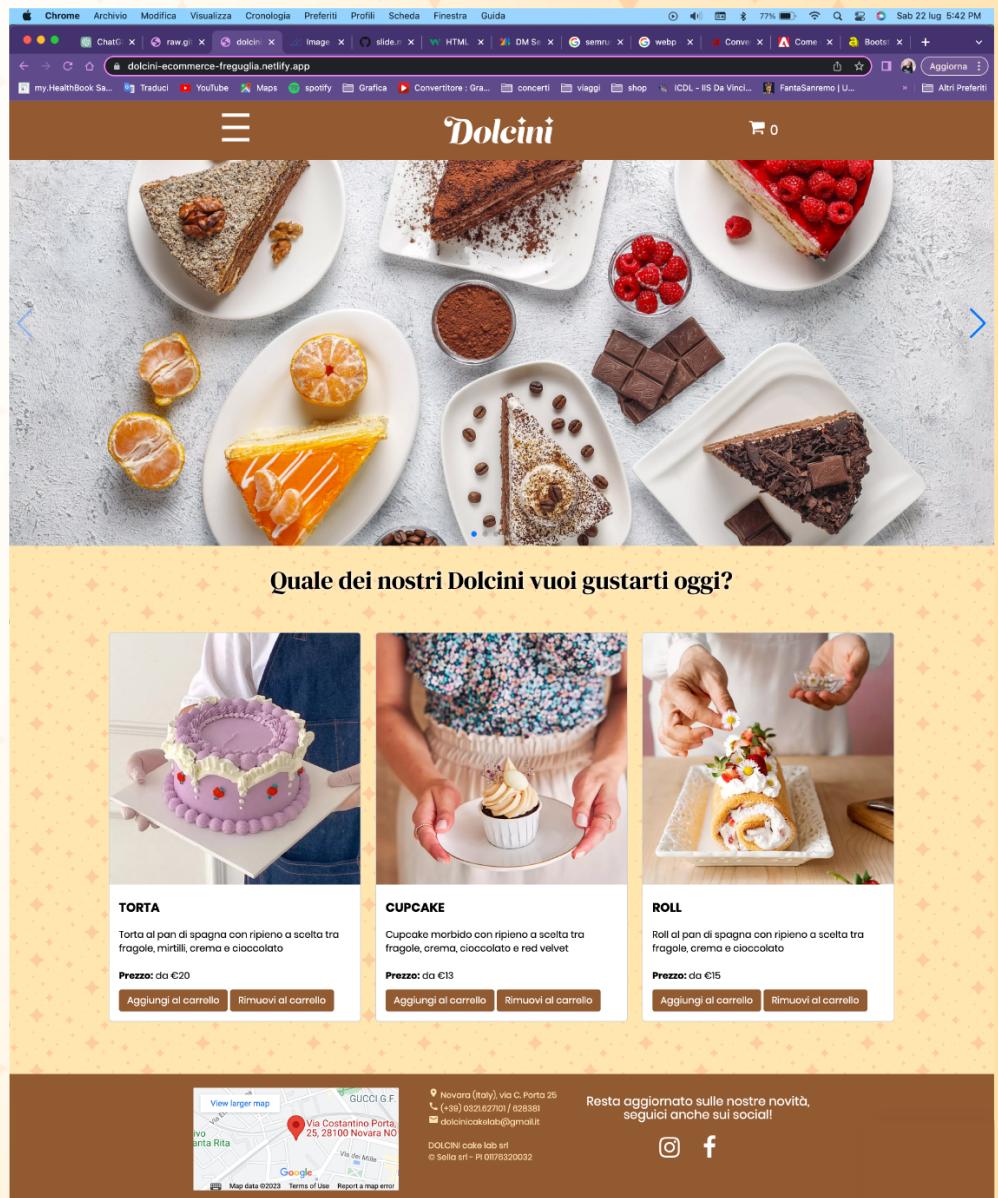
# Home page

Codice in js dello slider

```
/*swiper*/
var swiper = new Swiper(".mySwiper", {
    spaceBetween: 30,
    centeredSlides: true,
    autoplay: {
        delay: 2500,
        disableOnInteraction: false,
    },
    pagination: {
        el: ".swiper-pagination",
        clickable: true,
    },
    navigation: {
        nextEl: ".swiper-button-next",
        prevEl: ".swiper-button-prev",
    },
})
```

## Home page

- Sotto allo slider vi è un titolo che funziona da collegamento con un'ancora interna alla pagina, posizionata appena sopra, che permette al click sulla scritta di far salire la pagina, in modo da rendere visibili le tre caselle prodotto.

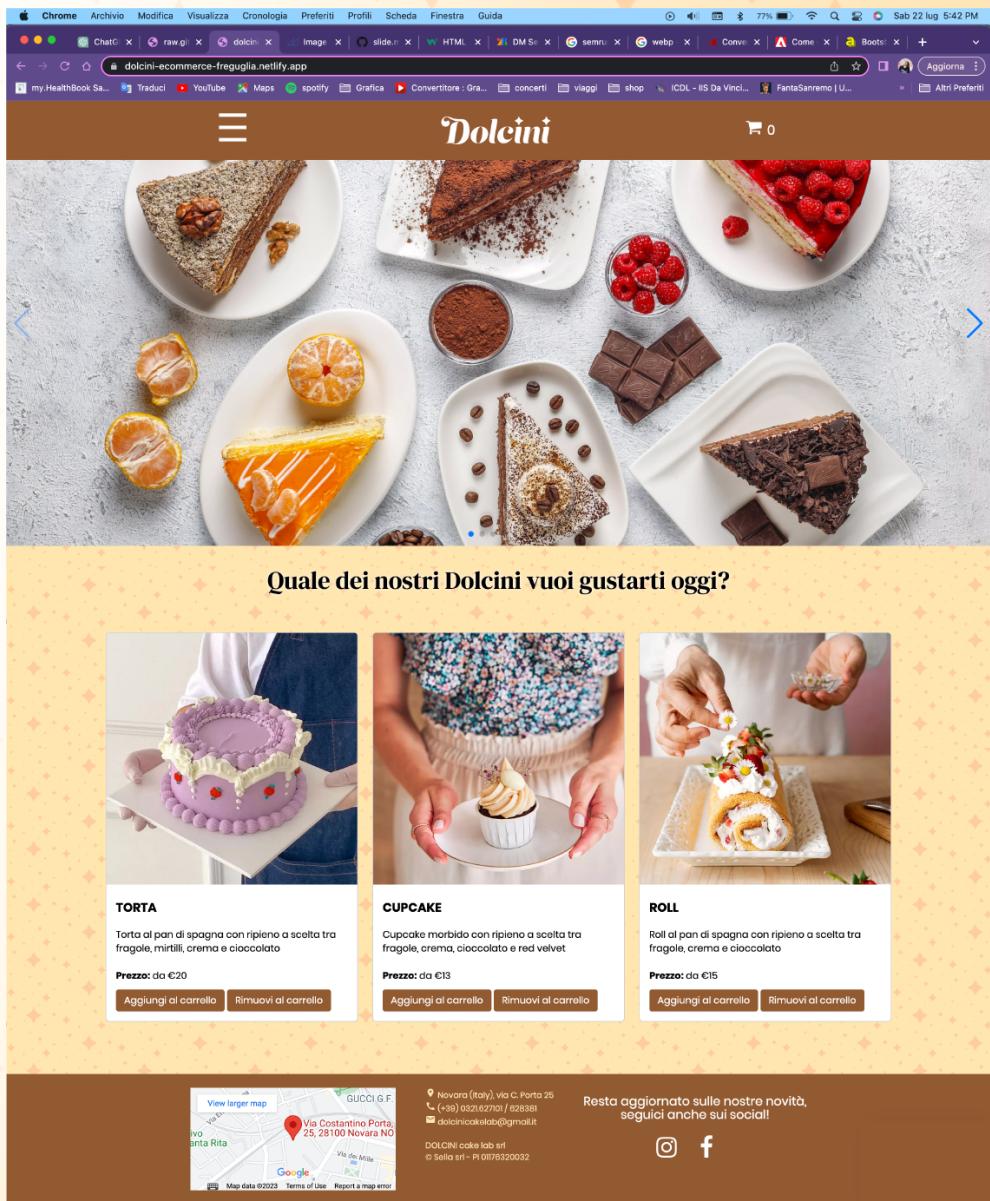


# Home page

- Sotto alla scitta ci sono le tre caselle prodotto che, usando la class 'card h-100', contengono l'immagine del prodotto, il suo nome, la descrizione, il costo e i tasti collegati al carrello per aggiungere o togliere quantità.

Al passaggio del puntatore sulle immagini dei prodotti, queste diminuiscono l'opacità e si crea un'ombra marrone attorno ad esse. Cliccandoci su invece, si viene reindirizzati alla pagine del prodotto stesso.

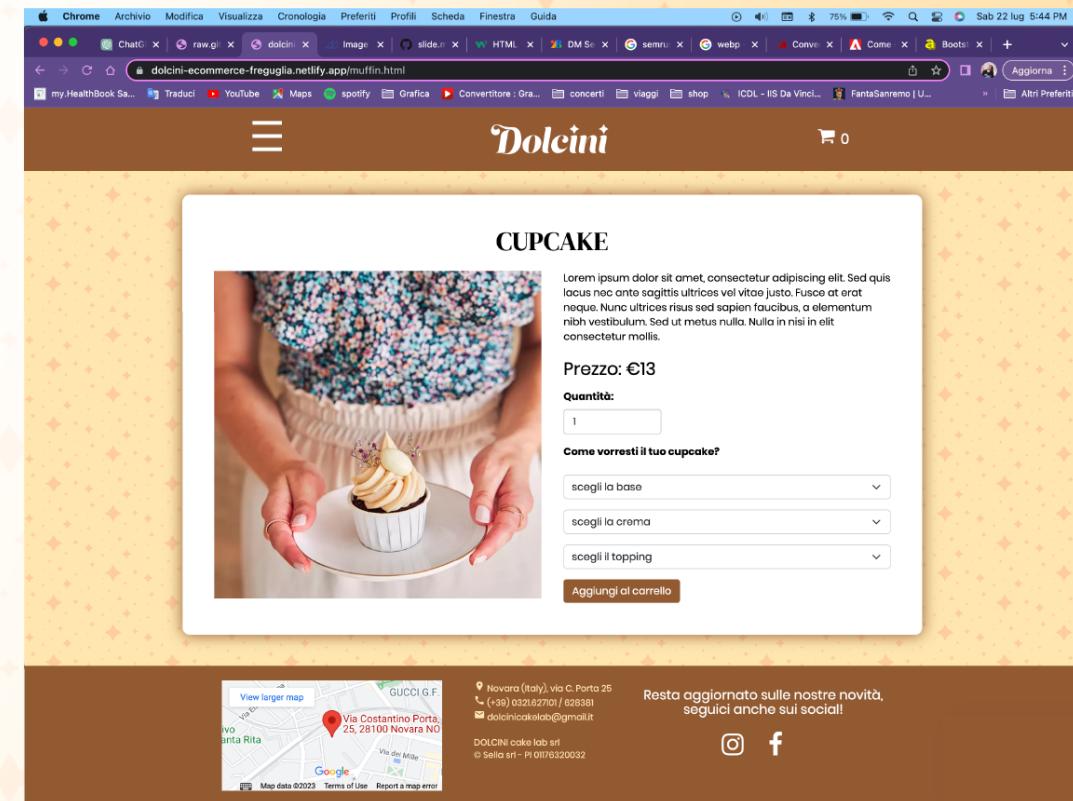
<https://github.com/ottafregu/ecommerce-freguglia-dolcini>



# Pagina prodotto

Nella pagina prodotto troviamo un container fluido di bootstrap, contenente due colonne in cui è inserita nuovamente la foto del prodotto e le informazioni a riguardo.

Il box si presenta bianco con un'ombra marrone all'esterno; nella parte delle informazioni sono presenti dei menù a tendina il cui codice nativo è stato preso da <https://getbootstrap.com/>, che permettono di scegliere quella che sarà la personalizzazione del prodotto.



# Uso dell'AI nel Carrello

ChatGPT mi ha aiutata nella creazione del codice per poter far funzionare i calcoli all'interno del carrello.

The screenshot shows a sequence of messages between a user and ChatGPT:

- User:** crea una stringa di codice in html e in javascript che mi permetta, aumentando o diminuendo la quantità di un prodotto che ha come costo 15€, di vedere il costo totale
- ChatGPT:** Certamente! Ecco un esempio di codice HTML e JavaScript che ti permette di aumentare o diminuire la quantità di un prodotto e visualizzare il costo totale in base al prezzo di 15€:

```
HTML:
<!DOCTYPE html>
<html>
<head>
    <title>Calcolatore Costo Totale</title>
</head>
<body>
    <h1>Calcolatore Costo Totale</h1>

    <label for="quantity">Quantità:</label>
    <input type="number" id="quantity" min="0" value="0">

    <p id="totalCost">Costo totale: €0</p>

    <script src="script.js"></script>

```
- User:** HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Calcolatore Costo Totale</title>
</head>
<body>
    <h1>Calcolatore Costo Totale</h1>

    <label for="quantity">Quantità:</label>
    <input type="number" id="quantity" min="0" value="0">

    <p id="totalCost">Costo totale: €0</p>

    <script src="script.js"></script>
```
- ChatGPT:** In questo esempio, abbiamo un campo di input numerico che rappresenta la quantità del prodotto. Ogni volta che viene inserito un valore nel campo, l'evento "input" viene attivato, e la funzione associata viene eseguita. La funzione recupera il valore della quantità inserita, moltiplica per il costo del prodotto (15€ nel nostro caso) e quindi aggiorna il testo del paragrafo con l'id "totalCost" per visualizzare il costo totale calcolato.
- User:** Assolutamente! Ecco la continuazione del codice HTML e JavaScript che ti permette di calcolare il costo totale di tre diversi prodotti e visualizzarlo:
- ChatGPT:** Torniamo alla mia prima richiesta, e consideriamo che non ho un prodotto solo ma 3 diversi prodotti con diversi costi. Inserisci già la funzione che mi hai detto per calcolare il costo totale di un prodotto, creami la prosecuzione di quella stringa in html e in javascript che mi permetta di calcolare il costo totale di tutto, sommando i costi totali dei 3 diversi prodotti
- ChatGPT:** La funzione 'updateCosts' viene richiamata ogni volta che viene inserito un valore nelle caselle di input delle quantità dei prodotti. La funzione calcola i costi totali per ciascun prodotto moltiplicando la quantità per il costo corrispondente. Quindi, aggiorna i paragrafi con gli id corrispondenti per visualizzare i costi totali di ciascun prodotto. Infine, calcola il totale generale sommando i costi totali dei tre prodotti e lo visualizza in un paragrafo separato.
- User:** ora, al totale generale, devo aggiungere 8€ della spedizione dei prodotti. Creami la funzione in html e javascript
- ChatGPT:** Spero che questo codice ti sia utile! Se hai ulteriori domande, non esitare a chiedere.
- ChatGPT:** Certamente! Ecco la funzione aggiornata in HTML e JavaScript per calcolare il costo totale dei prodotti, inclusi i 8€ di spese di spedizione:
- ChatGPT:** HTML:

```
<!DOCTYPE html>
<html>
<head>
    <title>Calcolatore Costo Totale</title>
</head>
<body>
    <h1>Calcolatore Costo Totale</h1>

    <label for="quantity">Quantità:</label>
    <input type="number" id="quantity" min="0" value="0">

    <p id="totalCost">Costo totale: €0</p>

    <script src="script.js"></script>
```
- ChatGPT:** totalCostOutput.textContent = "Costo totale: €" + totalCost;
totalCost2Output.textContent = "Costo totale: €" + totalCost2;
totalCost3Output.textContent = "Costo totale: €" + totalCost3;

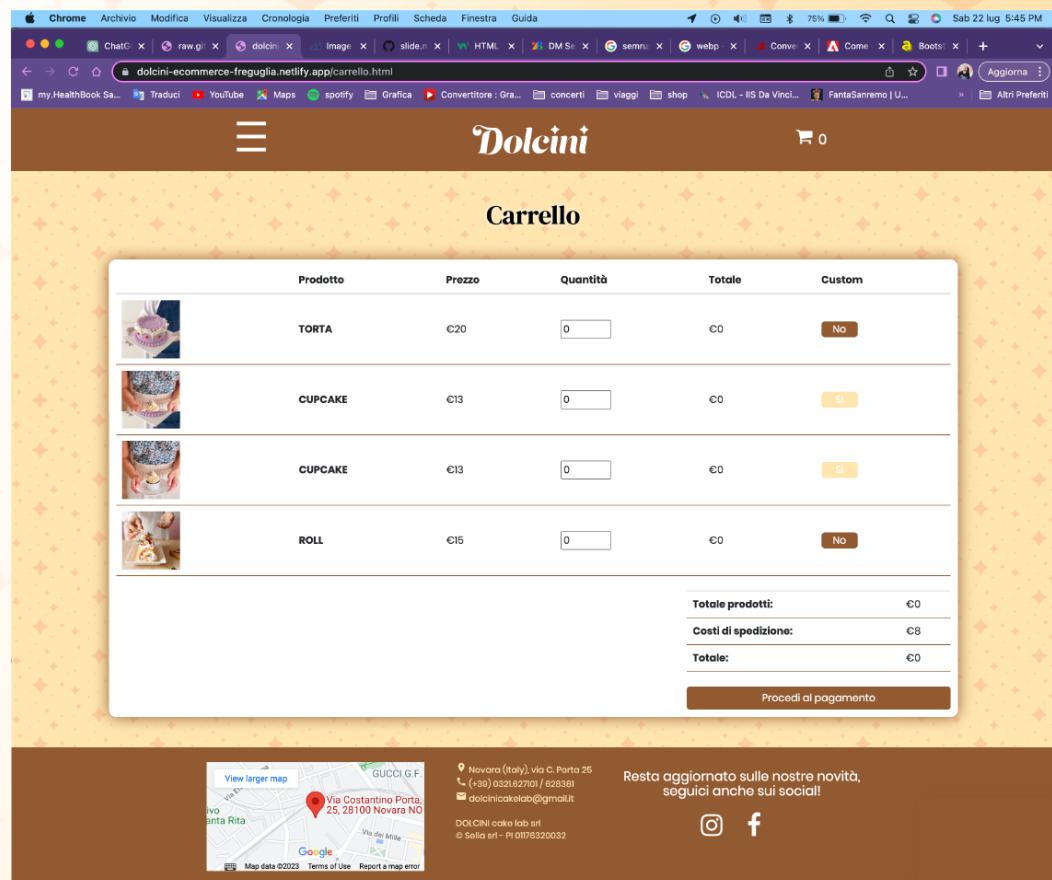
var subtotal = totalCost1 + totalCost2 + totalCost3;
var shippingCost = 8;
var grandTotal = subtotal + shippingCost;

subtotalOutput.textContent = "Totale parziale: €" + subtotal;
shippingCostOutput.textContent = "Spese di spedizione: €" + shippingCost;
grandTotalOutput.textContent = "Totale generale: €" + grandTotal;
- User:** In questo esempio, ho aggiunto un nuovo paragrafo per visualizzare il totale parziale ('`subtotal`') e un altro per le spese di spedizione ('`shippingCost`') impostato a 8€. Ho quindi aggiornato la funzione 'updateCosts' per calcolare il totale parziale sommando i costi totali dei prodotti e aggiungendo le spese di spedizione. Infine, ho aggiornato i paragrafi corrispondenti per visualizzare il totale parziale, le spese di spedizione e il totale generale.
- ChatGPT:** Spero che questa soluzione sia ciò che cercavi! Se hai ulteriori domande, non esitare a chiedere.

# Carrello

Cliccando nella parte di menù riguardante o nell'icona in alto a destra, veniamo rimandati alla pagina carrello. Qui troviamo sempre un container fluido come prima, ma con più righe e colonne.

Al suo interno sono riportati l'immagine del prodotto, il suo costo e la quantità che desideriamo (grazie ad una casella di testo editabile), il prezzo totale e la presenza o meno di personalizzazioni.

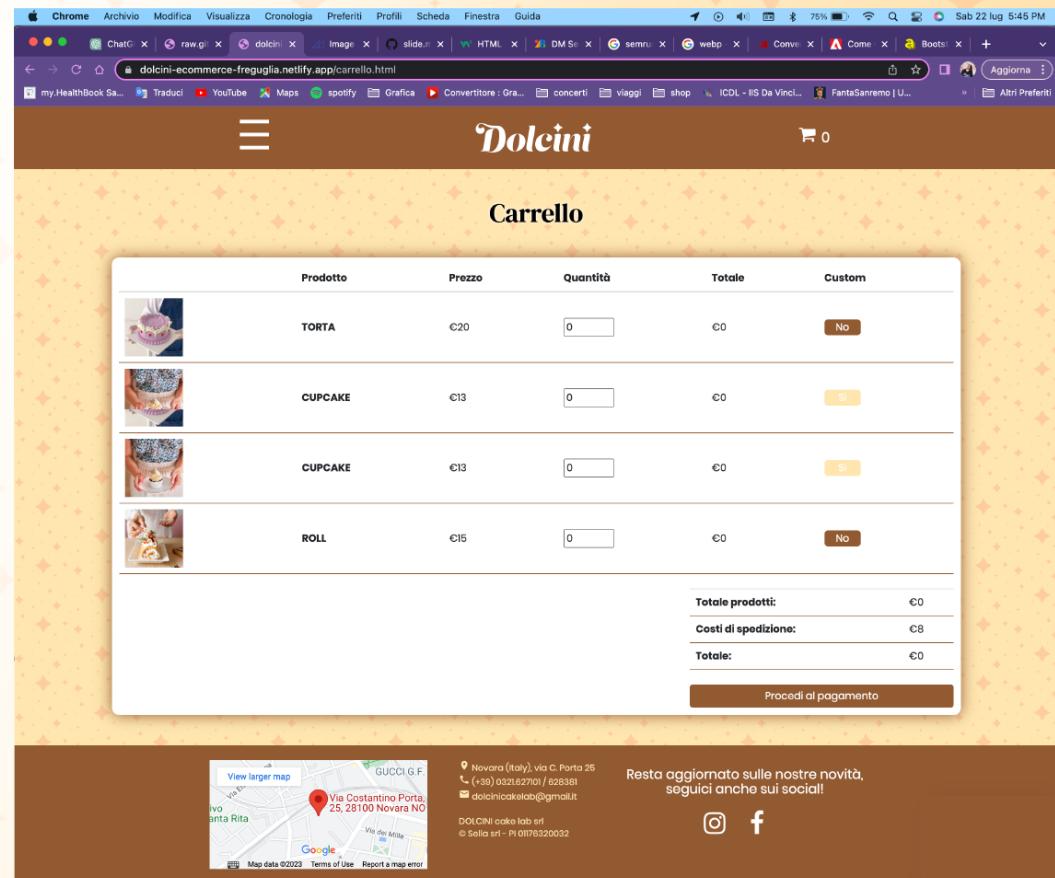


# Carrello

Aumentando la quantità di prodotto desiderato, si attiverà una funzione di calcolo 'updateCosts' che aggiornnerà il costo totale del prodotto moltiplicando il costo base per la quantità inserita.

Inoltre, sommerà i costi totali dei vari prodotti presenti nel carrello per avere un subtotale 'subTotal', visibile nella riga

**Totale prodotti**, per poi aggiungergli i costi di spedizione 'shippingCost' e aggiornare così il costo totale 'grandTotal' visibile nella riga **Totale**.



# Carrello

Codice in js del carrello

```
// prodotti
var quantity1Input = document.getElementById("quantity1");
var quantity2Input = document.getElementById("quantity2");
var quantity3Input = document.getElementById("quantity3");
var totalCost1Output = document.getElementById("totalCost1");
var totalCost2Output = document.getElementById("totalCost2");
var totalCost3Output = document.getElementById("totalCost3");
var subTotalOutput = document.getElementById("subTotal");
var shippingCostOutput = document.getElementById("shippingCost");
var grandTotalOutput = document.getElementById("grandTotal");

quantity1Input.addEventListener("input", updateCosts);
quantity2Input.addEventListener("input", updateCosts);
quantity3Input.addEventListener("input", updateCosts);

function updateCosts() {
    var quantity1 = parseInt(quantity1Input.value);
    var quantity2 = parseInt(quantity2Input.value);
    var quantity3 = parseInt(quantity3Input.value);

    var cost1 = 20;
    var cost2 = 13;
    var cost3 = 15;

    var totalCost1 = quantity1 * cost1;
    var totalCost2 = quantity2 * cost2;
    var totalCost3 = quantity3 * cost3;

    totalCost1Output.textContent = "€" + totalCost1;
    totalCost2Output.textContent = "€" + totalCost2;
    totalCost3Output.textContent = "€" + totalCost3;

    var subTotal = totalCost1 + totalCost2 + totalCost3;
    var shippingCost = 8;
    var grandTotal = subTotal + shippingCost;

    subTotalOutput.textContent = "€" + subTotal;
    shippingCostOutput.textContent = "€" + shippingCost;
    grandTotalOutput.textContent = "€" + grandTotal;
}
```

## About us

Il corpo della pagina ***chi siamo*** è diviso in 3 grandi fasce, riconoscibili dal distacco delle immagini di sfondo che ricoprono la larghezza dell'intera fascia:

- Nella prima troviamo un titolo in alto al centro e un box con testo posizionato basso al centro, grazie al tag "vertical-align: bottom"



## About us

- Nella seconda troviamo un box con opacità dell'80%, posizionato in alto a sinistra e contenente sia un titolo h2 che un testo
- Nella terza troviamo un box posizionato al centro e contenente del testo, con al di sotto un bottone di rimando alla homepage.

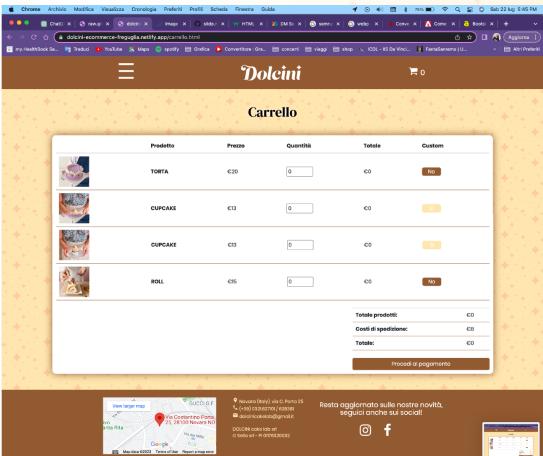
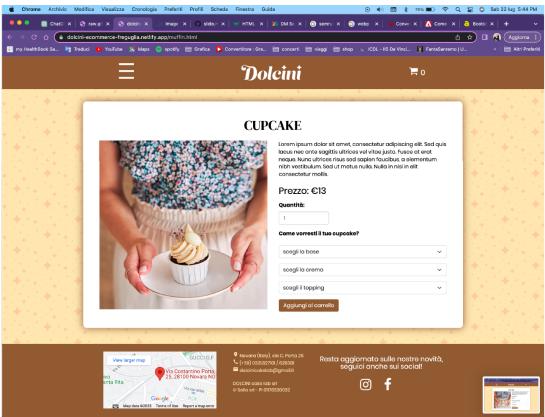
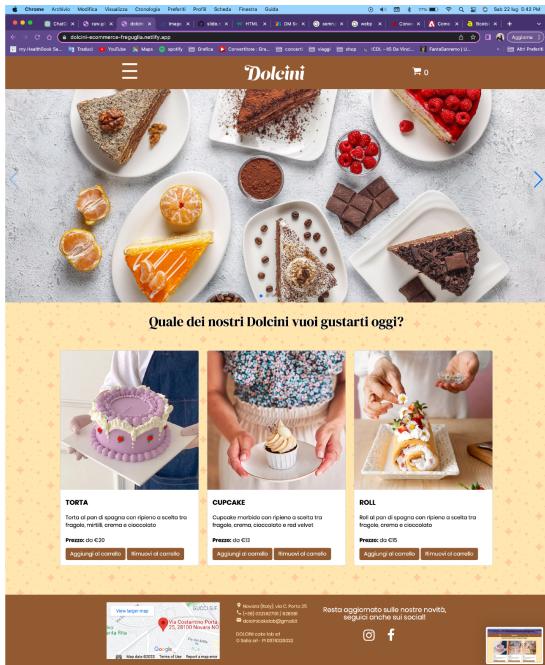


# Responsive delle pagine

*Adattamenti creati grazie alle **media query***

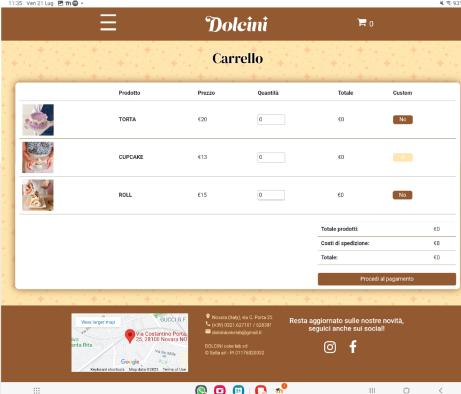
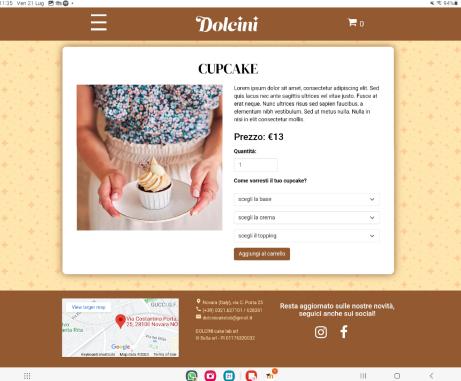
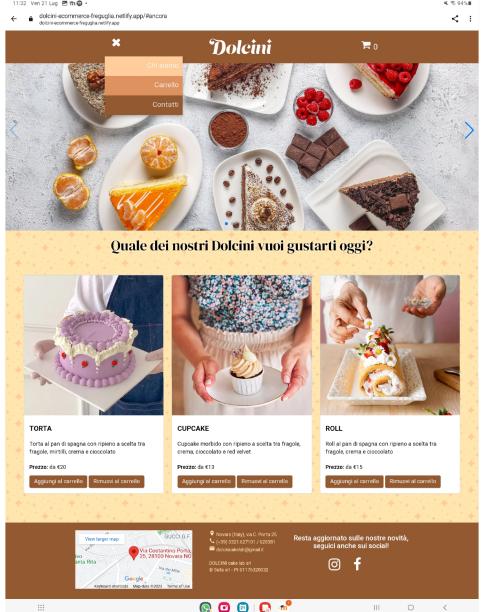
# SCHERMI MOLTO GRANDI

Più grandi di 1200px



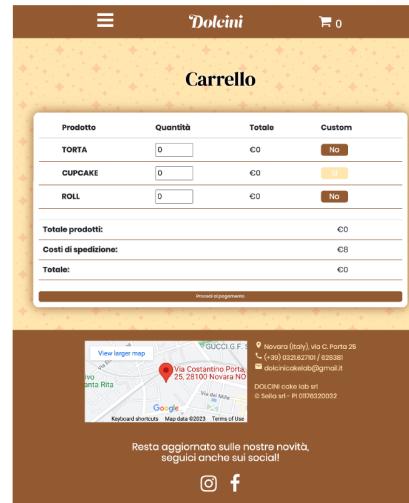
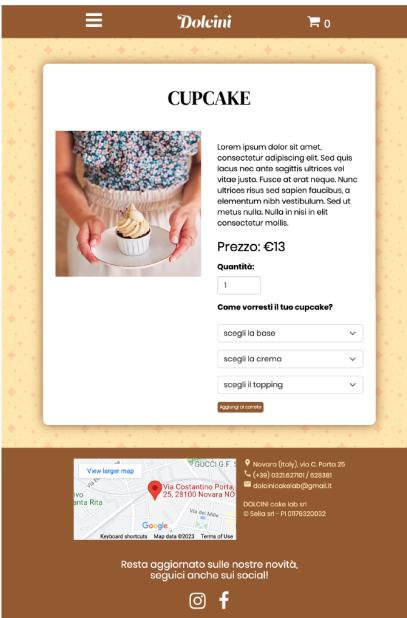
# SCHERMI GRANDI

Compresi tra 992px e 1200px



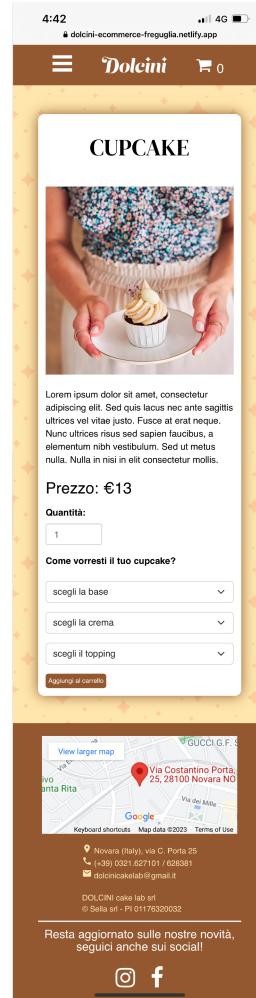
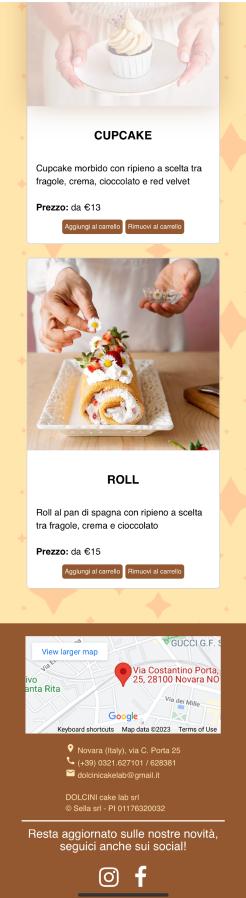
# SCHERMI MEDI

Compresi tra 768px e 992px



# SCHERMI MEDIO-PICCOLI

Compresi tra 576px e 768px



## Immagini

Tutte le immagini presenti in questo sito sono state convertite in formato **.webp** dal sito <https://anyconv.com/it/>.

Questo perchè il formato **.webp** è un **tipo di file per immagini raster sviluppato da Google** che utilizza un algoritmo di compressione avanzato, chiamato VP8 o VP9, che permette di ottenere **dimensioni di file più piccole** rispetto ad altri formati comunemente utilizzati, come il **.jpeg** o il **.png**, **mantenendo la qualità visiva dell'immagine.**

In sintesi, la scelta di utilizzare il formato **.webp** per le immagini del mio sito web mi ha consentito di **migliorare le prestazioni** complessive del sito e di offrire un'**esperienza più veloce e piacevole** ai visitatori.

# Prestazioni

PageSpeed Insights <https://pagespeed.web.dev/> è uno strumento gratuito di Google che permette di misurare la velocità di caricamento di una pagina web e le sue prestazioni.

