

Progetto di Applicazioni Web e Cloud: Fast Food

Realizzato da Martina Baga (martina.baga@studenti.unimi.it) e Emanuela Elli
(emanuela.elli@studenti.unimi.it)

Analisi dei requisiti

L'obiettivo del progetto è sviluppare un'applicazione web che implementi un sito di ordinazione online. Il servizio viene gestito sia lato ristoranti sia lato utenti. Inoltre abbiamo pensato di sviluppare un servizio volto a rispettare e a diffondere il rispetto dell'ambiente.

Identificazione delle funzionalità da sviluppare

Il sistema è composto da due macro scenari:

► gestione del **menù**

I menù vengono creati e gestiti dinamicamente, tramite salvataggio su localStorage, da parte di utenti ristoratori registrati sulla piattaforma. I ristoratori possono registrare la loro attività sul sito specificando: email e password dell'utente ristoratore, nome dell'attività, tipologia del ristorante, telefono per contattare il ristorante, partita IVA, indirizzo del locale (via, città, CAP). Inoltre ogni account ha la possibilità di fare il login ed il logout oppure di eliminare il proprio profilo.

Ogni ristoratore ha la possibilità di creare il proprio menù (primi, secondi, contorni, piatti unici, dolci) a partire da una lista di piatti generica disponibile per tutti gli utenti registrati ed ha la possibilità di modificare nome, prezzo, ingredienti per ogni piatto del menù. Inoltre è anche possibile poter aggiungere dei piatti completamente personalizzati all'interno del menù del ristorante specificando: il nome, la categoria (primo, secondo, contorno, piatto unico, dolce), il prezzo, gli ingredienti (vi è la possibilità di aggiungere ingredienti a partire da una lista comune ed anche la possibilità di aggiungere degli ingredienti manualmente).

► gestione degli **ordini**

Nel localStorage dei clienti sono memorizzate le seguenti informazioni: email, password, nome, cognome, indirizzo (via, città e CAP), il metodo di pagamento (pagamento alla consegna oppure tramite carta di credito, di cui sono memorizzati numero, scadenza e CVV2) e le preferenze. Queste informazioni possono essere modificate in qualunque momento dall'utente.

Il cliente dopo essersi registrato o autenticato può effettuare un ordine tra le 7 e le 23 presso un ristorante a sua scelta. Dopo aver scelto uno tra i ristoranti presenti nel localStorage, può aggiungere al suo ordine uno o più piatti tra quelli disponibili nel menù del ristorante scelto. Se il cliente ha inserito almeno un piatto nel riepilogo dell'ordine e se ha selezionato la tipologia di consegna (consegna a domicilio o ritiro al ristorante) allora può confermare

l'ordine. Quando l'utente ha completato l'ordine viene segnalato tramite un alert il tempo di elaborazione se ha scelto come metodo di consegna il ritiro presso il ristorante, mentre se ha selezionato la consegna a domicilio viene ridirezionato su un'altra schermata in cui viene specificato il costo delle consegna e le tempistiche necessarie per il ritiro dell'ordine.

Se la distanza tra il domicilio del cliente e l'indirizzo del ristorante supera i 50 km l'ordine viene automaticamente cancellato.

Il cliente può rivedere tutti i suoi ordini passati nella cronologia e nella pagina dedicata alla visualizzazione di queste informazioni può scegliere di cancellare un ordine dalla cronologia oppure di recensire il ristorante presso cui ha effettuato l'ordine.

Progettazione della struttura e della presentazione delle pagine web

Per la creazione del sito web ci siamo servite del framework Bootstrap, per cercare di creare una piattaforma responsive, oltre a delle pagine di stile create da noi per un'ulteriore personalizzazione. La struttura del progetto è la seguente:

```
all_in_green/
|   └── css/
|       |   └── style.css
|       └── stylePresentazione.css
|   └── html/
|       └── pagine html
|   └── img/
|       └── immagini
|   └── json/
|       |   └── piatti.json
|       └── ingredientiPiatti.json
└── relazione progetto
```

I file `piatti.json` e `ingredientiPiatti.json` contengono rispettivamente la lista dei piatti che hanno in comune tutti i ristoratori, e che i ristoratori possono aggiungere ai propri menù, e la lista degli ingredienti comuni che ogni ristoratore può scegliere per modificare un piatto già esistente oppure per creare un nuovo piatto personalizzato disponibile solo per il proprio ristorante. Di seguito vediamo due esempi, presi dai relativi file: il primo è l'esempio di un piatto facente parte della lista di piatti disponibile nella pagina di modifica del menù del ristorante, e sotto l'esempio di una categoria di ingredienti a disposizione per il ristoratore.

```
piatti = [
    {
        "id": 0,
        "nome": "Pizza",
        "categoria": "Piatto unico",
        "foto": "https://d1e3z2jco40k3v.cloudfront.net/-/...",
        "prezzo": 4.50,
```

```

    "ingredienti": ["acqua", "lievito", "sugo di pomodoro",
    "mozzarella", "basilico", "olio", "farina"]
}, ...
]

ingredientiPiatti = [
{
"categoria": "Ingredienti base",
"elementi": ["acqua", "farina", "lievito", "fecola di patate"]
}, ...
]

```

Le pagine html sono:

homepage.html	Contiene una breve descrizione del servizio.
presentazione.html	Contiene le informazioni relative al servizio, in particolare l'aspetto "green".

Lato ristorante:

registrazioneR.html	Contiene il form per la registrazione dei ristoratori al servizio.
loginR.html	Contiene il form per il login dei ristoratori al servizio.
homeR.html	Contiene la home del ristoratore con un riepilogo del menù, dei dati personali e la possibilità di uscire dal servizio o dell'eliminazione dalla piattaforma.
modificaDatiR.html	Contiene il form per la modifica del profilo del ristoratore e della sua attività.
creaPiatto.html	Contiene il form per la creazione di un piatto personalizzato.
modificaMenu.html	Contiene la lista generale dei piatti che si possono aggiungere al menù e la visualizzazione del riepilogo del menù con la possibilità di eliminare o modificare ogni piatto.
modificaPiatto.html	Contiene il form per la modifica di un piatto del menù del ristorante.
confermaOrdiniR.html	Contiene la visualizzazione degli ordini effettuati dal cliente, in modo tale che il ristoratore possa leggerli e provvedere alla preparazione e alla consegna. Una completato l'ordine, il ristoratore può eliminarlo cliccando sul bottone apposito.
statisticheR.html	Contiene le statistiche del numero complessivo di vendite settimanali e mensili avvenute nell'anno corrente effettuate dal ristorante loggato.
viewRecensioniR.html	Contiene la visualizzazione delle recensioni fatte lato cliente.

	Il ristoratore può decidere di eliminare le recensioni se non le ritiene opportune.
--	---

Lato cliente:

registrazioneU.html	Contiene il form per la registrazione dei clienti al servizio.
loginU.html	Contiene il form per il login dei clienti al servizio.
homeU.html	Contiene la home del cliente strutturata in due parti: <ul style="list-style-type: none">• la parte sinistra che presenta<ul style="list-style-type: none">◦ il riepilogo delle preferenze scelte in fase di registrazione e modificabili in qualsiasi momento◦ l'elenco dei piatti suggeriti sulla base dell'ultimo ordine effettuato• la parte destra che presenta<ul style="list-style-type: none">◦ il riepilogo dei dati personali modificabili in qualsiasi momento◦ la possibilità di uscire dall'account o di cancellarlo dalla piattaforma
modificaDatiU.html	Contiene il form per la modifica dei dati personali del cliente.
modificaPreferenzeU.html	Contiene il form per la modifica delle preferenze dell'utente.
scegliRistoranteU.html	Contiene l'elenco dei ristoranti permettendo di effettuare una ricerca per tipologia o per nome e di sceglierne uno per proseguire con l'ordine.
prenotazioneU.html	La pagina è strutturata in due parti: <ul style="list-style-type: none">• la parte sinistra presenta<ul style="list-style-type: none">◦ il riepilogo dei dati del ristorante e le sue recensioni◦ l'elenco dei piatti presenti nel menù del ristorante organizzati per tipologia (primo, secondo, contorno, piatto unico, dolce)◦ il form per effettuare una ricerca dei piatti per nome o per prezzo• la parte destra presenta il riepilogo dell'ordine contenente<ul style="list-style-type: none">◦ l'elenco dei piatti aggiunti all'ordine e per ciascuno di essi è indicato il prezzo del singolo piatto, la quantità e la possibilità di rimuovere un piatto dall'ordine stesso◦ il metodo di pagamento scelto dall'utente in fase di registrazione e modificabile successivamente◦ a scelta del metodo di consegna (consegna a domicilio o ritiro al ristorante)
ordiniU.html	Contiene la cronologia degli ordini dell'utente (cancellabili dalla cronologia) e per ciascuno di essi è possibile scrivere una o più recensioni.
recensioneU.html	Contiene il form per la scrittura della recensione.

mappa.html	Contiene le tempistiche di consegna oltre alla visualizzazione sulla mappa del ristorante a cui si è fatto l'ordine.
------------	--

La cartella `/img/` contiene le immagini che abbiamo caricato in locale all'interno delle pagine html.

Progettazione della sorgente di informazioni statica o dinamica

I macro-scenari presenti nel progetto sono due:

- ristoranti
- clienti

Le rispettive strutture dati vengono create (inizialmente vuote) nel localStorage al momento del primo accesso al sito web. Successivamente verranno modificate in modo dinamico a seconda delle azioni effettuate dal ristoratore o dal cliente (per esempio modifica del menù lato ristorante o modifica delle preferenze lato cliente).

La struttura dati presente nel localStorage dei **ristoratori** è la seguente:

- `l` → indica se il ristoratore è loggato oppure no
- `e` → indica l'email del ristoratore
- `p` → indica la password del ristoratore (nella home viene visualizzata con una serie di otto “*”)
- `nome` → indica il nome del ristorante
- `tipo` → indica la tipologia del ristorante (fitfood, cucina orientale o hamburgeria)
- `telefono` → indica il numero di telefono del ristorante
- `iva` → indica la partita IVA del ristoratore
- `indirizzo` → indica l'indirizzo in cui si trova il ristorante
- `città` → indica la città in cui si trova il ristorante
- `cap` → indica il CAP della città in cui si trova il ristorante
- `menu` → è un array di oggetti, ogni oggetto è composto da:
 - `nome` → indica il nome del piatto
 - `categoria` → indica la categoria del piatto (primo, secondo, etc)
 - `prezzo` → indica il prezzo del piatto
 - `foto` → indica la foto illustrativa del piatto
 - `ingredienti` → è un array composto dai nomi degli ingredienti di cui è composto il piatto
- `recensioni` → è un array di oggetti ognuno dei quali è composto da:
 - `nome` → indica il nome del cliente che ha scritto la recensione
 - `cognome` → indica il cognome del cliente che ha scritto la recensione
 - `valutazione` → indica la valutazione data dal cliente
 - `titolo` → indica il titolo della recensione
 - `commento` → indica il commento lasciato dal cliente
- `ordiniDaConfermare` → è un array di oggetti ognuno dei quali composto dalle seguenti informazioni:
 - `giorno` → indica il giorno in cui è stato effettuato l'ordine dal cliente

- mese → indica il mese in cui è stato effettuato l'ordine dal cliente
- anno → indica l'anno in cui è stato effettuato l'ordine dal cliente
- ora → indica l'ora in cui è stato effettuato l'ordine dal cliente
- minuti → indica i minuti in cui è stato effettuato l'ordine dal cliente
- piattiOrdine → indica la lista di piatti ordinati dal cliente
- emailCliente → indica l'email del cliente che ha effettuato l'ordine
- nomeCliente → indica il nome del cliente che ha effettuato l'ordine
- cognomeCliente → indica il cognome del cliente che ha effettuato l'ordine
- telefonoCliente → indica il numero di telefono del cliente che ha effettuato l'ordine
- indirizzoCliente → indica l'indirizzo di consegna del cliente che ha effettuato l'ordine
- cittaCliente → indica la città dell'indirizzo di consegna del cliente che ha effettuato l'ordine
- capCliente → indica il cap della città dov'è indicata la consegna
- metodoConsegnaCliente → indica il metodo di consegna scelto dal cliente (a domicilio o presso il ristorante)
- ordiniEffettuati → è un array uguale a ordiniDaConfermare che contiene gli ordini conclusi dal ristorante (quindi gli ordini vengono eliminati da ordiniDaConfermare e passati nell'array ordiniEffettuati)

Esempio:

```
[ {
  "l": 1,
  "e": "giorgio.rossi@gmail.com",
  "p": "E22B-hw",
  "nome": "Trattoria da Giorgio",
  "tipo": "hamburgeria",
  "telefono": 02314450,
  "iva": 86334519757,
  "indirizzo": "via Carducci 9",
  "citta": "Roma",
  "cap": 00100,
  "menu": [
    {
      nome: "fiorentina",
      categoria: "secondo",
      prezzo: 40,
      foto: "https://d1e3z2jco40k3v.cloudfront.net/-/...",
      ingredienti: ["carne chianina", "olio", "rosmarino"]
    }, ...
  ],
  "recensioni": [
    {
      nome: "Mario",
    }
  ]
}]
```

```

cognome: "Rossi",
valutazione: 4,
titolo: "molto buono",
commento: "cottura dei cibi perfetta"
}, ...

],
"ordiniDaConfermare": [
{
giorno: 12,
mese: 12,
anno: 2020,
ora: 13,
minuti: 20,
piattiOrdine: [["Pasta al pesto", 4]],
emailCliente: "giovanni.rossi@gmail.com",
nomeCliente: "giovanni",
cognomeCliente: "rossi",
telefonoCliente: "02314325",
indirizzoCliente: "via Guido Rossi 12",
cittaCliente: "Milano",
capCliente: "20090",
metodoConsegnaCliente: "consegna a domicilio"
}, ...
],
"ordiniEffettuati": [
{
giorno: 15,
mese: 10,
anno: 2020,
ora: 13,
minuti: 20,
piattiOrdine: [["Pasta al pesto", 1], ["Muffin ai mirtilli", 1]],
emailCliente: "sofia.rossi@gmail.com",
nomeCliente: "sofia",
cognomeCliente: "rossi",
telefonoCliente: "3324222444",
indirizzoCliente: "via Inganni 12",
cittaCliente: "Milano",
capCliente: "20090",
metodoConsegnaCliente: "consegna a domicilio"
}, ...
], ...
]
}

```

La struttura dati presente nel localStorage dei **clienti** è la seguente:

- l → indica se il cliente è loggato oppure no
- nc → indica l'email del cliente
- p → indica la password del cliente (nella home viene visualizzata con una serie di otto “ * ”)
- nome → indica il nome del cliente
- cognome → indica il cognome del cliente
- indirizzo → indica l'indirizzo del cliente
- citta → indica la città in cui vive il cliente
- cap → indica il CAP della città in cui vive il cliente
- telefono → indica il numero di telefono del cliente
- fitfood → indica la partita preferenza dell'utente se il valore è “1”
- hamburgeria → indica la partita preferenza dell'utente se il valore è “1”
- cucinaorientale → indica la partita preferenza dell'utente se il valore è “1”
- pagamento → indica il metodo di pagamento del cliente (“pagamento alla consegna” oppure “carta di credito”)
- numerocarta → indica il numero di carta di credito del cliente se il metodo di pagamento è “carta di credito” altrimenti vale “”
- mesescadenza → indica il mese di scadenza della carta di credito del cliente se il metodo di pagamento è “carta di credito” altrimenti vale “”
- annoscadenza → indica l'anno di scadenza della carta di credito del cliente se il metodo di pagamento è “carta di credito” altrimenti vale “”
- cvv2 → indica il CVV2 della carta di credito del cliente se il metodo di pagamento è “carta di credito” altrimenti vale “”
- prenotazione → è un oggetto composto da:
 - giorno → indica il giorno in cui il cliente effettua l'ordine
 - mese → indica il mese in cui il cliente effettua l'ordine
 - anno → indica l'anno in cui il cliente effettua l'ordine
 - ora → indica l'ora in cui il cliente effettua l'ordine
 - minuti → indica i minuti in cui il cliente effettua l'ordine
 - ristorante → indica l'email del ristorante presso cui il cliente effettua l'ordine (si suppone infatti che l'email del ristorante sia univoca)
 - piattiOrdine → è un array bidimensionale in cui piattiOrdine[0] contiene il nome del piatto e piattiOrdine[1] ne contiene la quantità ordinata dal cliente
 - totale → indica il totale dell'ordine effettuato dal cliente
 - pagamento → indica il metodo di pagamento registrato nel localStorage del cliente
 - consegna → indica il metodo di consegna scelto dal cliente (“consegna a domicilio” oppure “ritiro presso il ristorante”)
 - tempoelaborazione → indica il tempo necessario al ristorante per preparare l'ordine del cliente (si stimano all'incirca 7 minuti per piatto) valutando la quantità dei piatti presente nel suo ordine e la quantità dei piatti negli ordini ancora in fase di preparazione del ristorante

Esempio:

```
[ {  
    "l":1,  
    "nc": "mario.rossi@gmail.com",  
    "p": "mariorossi",  
    "nome": "Mario",  
    "cognome": "Rossi",  
    "indirizzo": "via Celoria 18",  
    "città": "Milano",  
    "cap": "20133",  
    "telefono": "0245635589",  
    "fitfood": "1",  
    "hamburgeria": "0",  
    "cucinaorientale": "1",  
    "pagamento": "pagamento alla consegna",  
    "numerocarta": "",  
    "mesescadenza": "",  
    "annoscadenza": "",  
    "cvv2": "",  
    "prenotazione": [  
        {  
            giorno: "14",  
            mese: "1",  
            anno: "2021",  
            ora: "10",  
            minuti: "15",  
            ristorante: "giorgio.rossi@gmail.com",  
            piattiOrdine: [["Pasta al pesto", 2]],  
            totale: 12,  
            pagamento: "pagamento alla consegna",  
            consegna: "ritiro presso il ristorante",  
            tempoelaborazione: 14  
        }, ...  
    ], ...  
} ]
```

Implementazione pagine web

Ogni pagina è composta da una navbar differente a seconda che l'utente si trovi in una delle seguenti situazioni:

- l'utente non si è ancora autenticato/registrato



Home Chi siamo Registrati qui! ▾ Accedi ▾

- l'utente si è registrato e autenticato come ristoratore



All in green

Home Recensioni Conferma gli ordini Statistiche vendite

- l'utente si è registrato e autenticato come cliente



All in green

Home Ordina ora Cronologia degli ordini

Di seguito vi è la descrizione delle pagine web create per lo sviluppo di questo servizio.

homepage.html e presentazione.html

Sono due pagine create per accogliere un qualsiasi utente (che sia già registrato al servizio oppure no) e in cui spieghiamo quale tipologia di servizio offre All In Green e quali sono le sue politiche.

registrazioneR.html, registrazioneU.html, loginR.html e loginU.html

Tramite queste pagine permettiamo all'utente di registrarsi o accedere alla piattaforma come cliente oppure come ristoratore.

Entrambe le pagine di registrazione presentano le seguenti funzioni Javascript:

- inizializzazioneStorage() per verificare l'esistenza di un localStorage contenente i clienti oppure i ristoratori, nel caso in cui non sia ancora presente ne crea uno vuoto

```
if (typeof(localStorage.utenti) == "undefined") {  
    localStorage.utenti="[]";  
}
```

- checkLogin() per caricare il localStorage dei clienti oppure dei ristoratori
- uguali() per verificare che non si registrino due clienti oppure due ristoratori con la stessa email

```
function uguali(vristoranti,datiform) {  
    if (vristoranti.e==datiform.e)  
        return true;  
    return false;  
}
```

- `registrazione()` per permettere l'effettiva registrazione del cliente oppure del ristoratore mediante la costruzione di un oggetto composto dai dati relativi ai campi del form

```

var datiform = {
    l:1,
    e:document.registrazione.email.value,
    p:document.registrazione.password.value,
    nome:document.registrazione.nome.value,
    tipo:document.registrazione.tipo.value,
    telefono:document.registrazione.telefono.value,
    iva:document.registrazione.iva.value,
    indirizzo:document.registrazione.indirizzo.value,
    citta:document.registrazione.citta.value,
    cap:document.registrazione.cap.value,
    menu: [],
    recensioni: [],
    ordiniDaConfermare: [],
    ordiniEffettuati: []
}

```

Dopo i relativi controlli sull'esistenza di un utente con la stessa email e dell'effettiva compilazione di tutti i campi necessari (in entrambi i casi viene inviato un alert), l'oggetto creato viene inserito nel localStorage e l'utente viene reindirizzato alla propria pagina home.

```

vristoranti[vristoranti.length] = datiform;
localStorage.ristoranti = JSON.stringify(vristoranti);
window.location.href = "homeR.html";
return true;

```

homeR.html e homeU.html

Entrambe le pagine stampano dinamicamente i dati del profilo dell'utente tramite una funzione Javascript che analizza l'utente che si è loggato nel sistema e ne recupera le informazioni dal localStorage caricato nel seguente modo:

```

vristoranti = JSON.parse(localStorage.getItem("ristoranti"))
vutenti = JSON.parse(localStorage.getItem("utenti"))

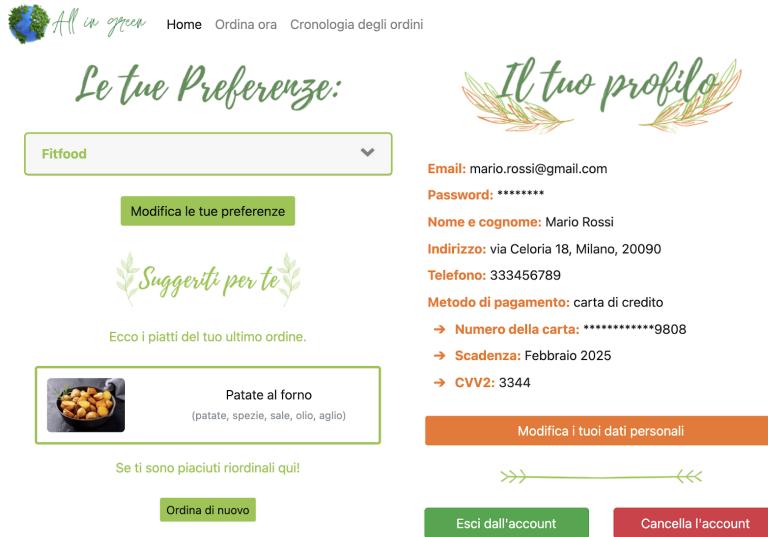
```

Sempre attraverso questo meccanismo estraiamo le informazioni riguardanti il menù del ristorante o le preferenze (ed eventuali suggerimenti del cliente).

homeR.html



homeU.html



modificaDatiR.html, modificaDatiU.html e modificaPreferenzeU.html

Entrambe le pagine presentano un form per la modifica dei dati personali dell'utente, ovvero quelli che ha inserito nel form in fase di registrazione. In particolare attraverso la funzione Javascript `getFromUrl()` ricaviamo con un ciclo `for` la posizione dell'utente all'interno dell'array di tutti i ristoratori oppure clienti nel `localStorage`. Con la funzione `modificaDati()` controlliamo che:

- email, partita IVA, numero di telefono dei ristoratori e email e numeri di telefono dei clienti siano univoci

```
for(i=0;i<vristoranti.length;i++) {
```

```

        if (i != posizione) {
            if(vristoranti[i].e==document.modifica.e.value) {
                alert("ERRORE: questa email è già registrata,
                      sceglie un'altra!");
                return false;
            }
            ...
        }
    }

```

- gli input inseriti nei campi non siano uguali ai dati già esistenti nel localStorage

```

if (vristoranti[posizione].e==document.modifica.email.value) {
    alert("ERRORE: questa è la tua vecchia email!");
    return false;
}

```

- se i campi risultano vuoti vengono salvati nuovamente quelli precedenti

```

} else if (document.modifica.email.value == "") {
    vristoranti[posizione].e = vristoranti[posizione].e;
}

```

- altrimenti salva le nuove informazioni

```

} else {
    vristoranti[posizione].e = document.modifica.email.value;
}

```

Un procedimento analogo avviene per la modifica delle preferenze del cliente: a seconda che una box sia checked oppure no (quindi che il suo valore sia “1” o “0”) il valore di quella preferenza nel localStorage del cliente viene modificato.

```

vutenti[posizione].fitfood=document.modifica.fitfood.value;
vutenti[posizione].hamburgeria=document.modifica.hamburgeria.v
alue;
vutenti[posizione].cucinaorientale=document.modifica.cucinaori
entale.value;

```

creaPiatto.html, modificaMenu.html e modificaPiatto.html

Queste pagine servono lato ristoratore per poter modificare il proprio menù aggiungendo un piatto scelto da una lista comune a tutti i ristoranti (che è salvata all'interno dei file JSON piatti.json) oppure per creare un piatto personalizzato tramite la form all'interno della pagina `creaPiatto.html`. Infine l'utente può gestire i propri piatti, sia personalizzati che non, modificando il nome, il prezzo e gli ingredienti, e può eliminare i piatti all'interno del proprio menù.

Nella pagina `modificaMenu.html` facciamo stampare la lista dei piatti disponibile a tutti i ristoratori, a seconda della categoria che l'utente sceglie di visualizzare, attraverso la

funzione `stampaListaPiatti(nomeCategoria)`. Il “`nomeCategoria`” lo otteniamo tramite la funzione `onclick="stampaListaPiatti('...')"` all’interno dei bottoni che specificano le categorie da visualizzare. Una volta chiamata la funzione si recuperano i dati dei piatti dal documento `piatti.json`.

All’interno della stessa pagina, sulla destra dello schermo, facciamo visualizzare all’utente il riepilogo del proprio menù con la possibilità di gestire ognuno dei propri piatti tramite i bottoni:

- “Modifica piatto” che chiama la funzione `modificaPiatto(numero)` in cui viene passato il numero del piatto all’interno dell’array “menu” associata al ristorante che permette di cambiare schermata e compilare la form di modifica del piatto
- “Cancella piatto” che chiama la funzione `cancellaPiatto(numero)` a cui viene passato il numero del piatto all’interno dell’array “menu” associato al ristorante e tramite le seguenti istruzioni

```
vristoranti[ristorante_loggato].menu.splice(numero,1);
localStorage.ristoranti = JSON.stringify(vristoranti);
```

eliminiamo il piatto dal menu del ristorante e salviamo i dati del localStorage

modificaMenu.html

The screenshot shows two side-by-side panels. The left panel has a green header with a fork icon and the text "Componi il tuo menu:". Below it is a button bar with "Primi", "Secondi", "Contorni", "Piatti unici", and "Dolci". Under "Primi", there's a section for "Pasta al pesto" with a list of ingredients (pasta, basilico, pinoli, grana padano, olio) and a "Aggiungi al tuo menu" button. To its right is a photo of the dish. The right panel has a green header with a floral wreath and the word "Menu:". It shows sections for "Primi" (with a dropdown for "Pasta al pesto" and buttons for "Modifica piatto" and "Cancella piatto dal menu"), "Secondi" (with a dropdown for "Insalata mista" and similar buttons), and "Contorni" (with a message "Nessun contorno in vendita:("). Both panels have a navigation bar at the top with "All in green", "Home", "Recensioni", "Conferma gli ordini", and "Statistiche vendite".

Nelle pagine di `modificaPiatto.html` e `creaPiatto.html` viene visualizzata una form che viene stampato tramite la funzione `stampaForm()` del Javascript in modo tale che si possano visualizzare gli ingredienti prelevandoli dal json (nel caso degli ingredienti aggiuntivi) o dal localStorgage (nel caso degli ingredienti del piatto) e permettendo al ristoratore di selezionare gli ingredienti che vuole aggiungere oppure di deselectingare gli ingredienti che sono già presenti nel suo piatto per poterli eliminare.

Il form viene stampato aggiungendo nell’html le seguenti istruzioni:

```
<body onload="stampaForm()">
```

```
<div id="form"></div>
```

Mentre nella funzione `stampaForm()` è presente:

```
var form = new String("");
document.getElementById("form").innerHTML = form;
```

Alla fine della form sono presenti i bottoni:

- modifica → chiama la funzione `modificaValori()`, nel caso di `modificaPiatto.html`, che provvederà a sostituire i valori dei campi compilati con i valori che erano salvati precedentemente, altrimenti mantiene quelli già salvati, oppure `aggiungiNuovoPiatto()` nel caso di `creaPiatto.html` in cui crea un nuovo oggetto (strutturato come i piatti salvati nell'array menu) e lo aggiunge all'array del menu
- pulisci form → permette di pulire tutti i campi del form
- annulla e torna indietro → permette di tornare alla pagina `modificaMenu.html` (nel caso di `modificaPiatto.html`) oppure di tornare alla pagina `homeR.html` (nel caso di `creaPiatto.html`)

modificaPiatto.html (la struttura della pagina è molto simile a creaPiatto.html)

The screenshot shows a web form titled "Modifica il tuo piatto:" with a decorative orange and yellow brushstroke background. The form fields include:

- Nome:** Input field containing "Insalata mista".
- Prezzo:** Input field containing "5 €".
- Ingredienti:** A list of checked checkboxes: latuga, olio, sale, aceto balsamico, rapanelli, carote, rucola.
- Ingredienti base:** A dropdown menu showing options: acqua, farina, lievito, fécule de patate.
- Spezie:** A dropdown menu.
- Condimento:** A dropdown menu.
- Puoi aggiungere altri ingredienti a tuo piacimento:** Four input fields labeled "Aggiungi un ingrediente a tua scelta".
- Buttons:** A green "Modifica" button and a blue "Pulisci form" button at the bottom.
- Footer:** A red button labeled "Annulla e torna alla modifica del menu."

scegliRistoranteU.html

In questa pagina il cliente può visualizzare l'elenco tutti i ristoranti presenti nel localStorage grazie alla funzione Javascript `stampaRistoranti()`. Inoltre può effettuare una ricerca dei ristoranti per nome (form con input di tipo testo) oppure per tipologia (form con input di tipo radio-button) e i risultati vengono individuati grazie alla funzione Javascript `ricercaRistoranti()`. Questa funzione inizialmente salva in due variabili i valori inseriti negli input.

```
nomeris = document.cercaristorante.nomeristorante;
var tiporis = new String("");
for (j=0; j<document.cercaristorante.tiporistorante.length; j++) {
    if (document.cercaristorante.tiporistorante[j].checked) {
        tiporis += document.cercaristorante.tiporistorante[j].value;
    }
}
```

Successivamente li confronta con le informazioni di tutti i ristoranti salvati nel localStorage. Quando un ristorante viene selezionato, il cliente viene indirizzato alla pagina `prenotazioneU.html` tramite la funzione Javascript `prenotaRistorante(i)` dove i è la posizione del ristorante selezionato all'interno del localStorage di tutti i ristoranti.

```
window.location.href="prenotazioneU.html?"+i;
```

Ricerca ristorante:

Nome del ristorante

Tipo di ristorante
 Fitfood
 Hamburgeria
 Cucina orientale

prenotazioneU.html

In questa pagina il cliente visualizza le informazioni relative al ristorante selezionato nella pagina precedente (cioè `scegliRistorante.html`) grazie alla funzione `getFromUrl()` che permette di ricavare dall'url la posizione del ristorante all'interno del localStorage e che salviamo nella variabile `idris`.

Grazie alla funzione Javascript `stampaDatiRistorante(idris)` il cliente può visualizzare i dati, le recensioni e il menù del ristorante suddiviso in 5 settori: primi, secondi, contorni, piatti unici e dolci.

Menù del ristorante:

<input type="text" value="Cerca un piatto"/>	<input type="button" value="Cerca"/> <input type="button" value="Annulla"/>
<input type="checkbox"/> ≤ 5€ <input type="checkbox"/> > 5€	

I piatti del ristorante

Primi	▼
Secondi	▼
Contorni	▼
Piatti unici	▼
Dolci	▼

In particolare nel menù è possibile effettuare due tipologie di ricerche:

- per nome, tramite le due funzioni Javascript `ricercaPiatto(idris)` e `svuotaRicercaPiatto()`

Menù del ristorante:

<input type="text" value="pasta"/>	<input type="button" value="Cerca"/> <input type="button" value="Annulla"/>
------------------------------------	---

Risultati della ricerca per nome

Pasta al pesto	6 euro	<input type="button" value="Aggiungi"/>
Ingredienti: pasta, basilico, pinoli, grana padano, olio		

- per prezzo, tramite le due funzioni Javascript `ricercaPiattoPrezzo(idris)` e `svuotaRicercaPiattoPrezzo()`

Menù del ristorante:

<input type="checkbox"/> ≤ 5€ <input checked="" type="checkbox"/> > 5€	<input type="button" value="Cerca"/> <input type="button" value="Annulla"/>
---	---

Risultati della ricerca per prezzo

Pasta al pesto	6 euro	<input type="button" value="Aggiungi"/>
Ingredienti: pasta, basilico, pinoli, grana padano, olio		
Vellutata di zucca	6.5 euro	<input type="button" value="Aggiungi"/>
Ingredienti: zucca, patate, pepe, prezzemolo, olio		

Le funzioni di ricerca sono analoghe in quanto confrontano i campi compilati dal cliente e i dati del menù del ristorante. In particolare per la ricerca per nome abbiamo inserito queste righe di codice per effettuare una ricerca più ampia del nome inserito:

```
var nomePiattoRistorante = vristoranti[idris].menu[j].nome;
nomePiattoRistorante = nomePiattoRistorante.toLowerCase();
var ricercaParola = nomePiattoRistorante.match(nomePiattoRicerca);
```

Anche le le due funzioni di svuotamento della ricerca sono analoghe in quanto svuotano i campi testo o radio-button compilati e mostrano nuovamente il menù completo.

Quando un piatto viene aggiunto al riepilogo dell'ordine tramite il bottone Aggiungi, la funzione Javascript riepilogoOrdine(idris, idpiatto), che prende in input la posizione del ristorante all'interno del localStorage e la posizione del piatto (da aggiungere all'ordine del cliente) all'interno dell'array menu del ristorante, esegue una serie di operazioni:

- aggiorna il totale dell'ordine

```
totaleOrdine += vristoranti[idris].menu[idpiatto].prezzo;
```

- aggiunge un nuovo piatto (un array in cui in posizione 0 si trova il nome del piatto e in posizione 1 la sua quantità) nell'array piattiOrdine nel caso in cui non ve ne sia ancora uno

```
if (piattiOrdine.length==0) {  
    var nuovoPiattoArray = new Array();  
    nuovoPiattoArray[0] = nomePiattoSelezionato;  
    nuovoPiattoArray[1] = 1;  
    piattiOrdine.push(nuovoPiattoArray);  
}
```

- aggiorna la quantità del piatto nell'array piattiOrdine nel caso in cui il piatto sia già stato aggiunto in precedenza

```
var count = 0;  
for (i=0; i<piattiOrdine.length; i++) {  
    if (piattiOrdine[i][0]== nomePiattoSelezionato) {  
        piattiOrdine[i][1]+= 1;  
        count += 1;  
        quantitaPiattoSelezionato = piattiOrdine[i][1];  
    }  
}
```

- aggiunge un nuovo piatto nell'array piattiOrdine nel caso in cui il piatto non sia ancora stato aggiunto in precedenza

```
if (count==0) {  
    var nuovoPiattoArray = new Array();  
    nuovoPiattoArray[0] = nomePiattoSelezionato;  
    nuovoPiattoArray[1] = 1;  
    piattiOrdine.push(nuovoPiattoArray);  
}
```

- stampa il riepilogo aggiornato, il metodo di pagamento scelto dall'utente e il form per la scelta del metodo di consegna

Riepilogo dell'ordine:

Il ristorante è aperto dalle ore 7 alle ore 23.

Piatto	Quantità	Prezzo singolo	
Insalata mista	1	5 euro	<button>Rimuovi</button>

Totale: 5 euro

Metodo di pagamento

Pagamento alla consegna

Scegli il metodo di consegna

Consegna a domicilio (via pace 21, milano, 20122)

I costi per la consegna sono di 2€ per chilometro percorso.

Ritiro presso il ristorante (via giovanni celoria 18, milano, 20133)

Conferma ordine

Nel riepilogo dell'ordine il cliente può scegliere di rimuovere i piatti aggiunti tramite il bottone Rimuovi che richiama funzione Javascript `rimuoviPiatto(idris, idpiattoS, posizionePS)`, che prende in input la posizione del ristorante all'interno del localStorage, la posizione del piatto (da rimuovere dall'ordine del cliente) all'interno dell'array menu del ristorante e la posizione del piatto (da rimuovere dall'ordine del cliente) all'interno dell'array piattiOrdine. Questa funzione esegue una serie di operazioni:

- sottrae uno dalla quantità del piatto all'interno dell'array piattiOrdine e se la quantità del piatto era già pari a 1 allora il piatto viene rimosso dall'array piattiOrdine

```
if (piattoSelezionatoArray[1] > 1) {
    piattoSelezionatoArray[1] = piattoSelezionatoArray[1]-1;
} else {
    piattiOrdine.splice(posizionePS, 1);
}
```

- aggiorna il totale dell'ordine

```
totaleOrdine -= vristoranti[idris].menu[idpiattoS].prezzo;
```

- stampa il riepilogo aggiornato, il metodo di pagamento scelto dall'utente e il form per la scelta del metodo di consegna

Nel riepilogo dell'ordine il cliente può confermare l'ordine stesso tramite il bottone Conferma ordine (il bottone è visibile solo se il cliente ha inserito almeno un piatto nel suo ordine) che richiama la funzione Javascript `confermaOrdine(idris)`, che prende in input la posizione del ristorante all'interno del localStorage. La funzione esegue una serie di operazioni:

- carica il localStorage dei clienti e dei ristoranti, individuando quale cliente sta completando l'ordine

```
var vristoranti = JSON.parse(localStorage.ristoranti);
var vutenti = JSON.parse(localStorage.utenti);
var utente_loggato = 0;
```

```

for (var i=0; i<vutenti.length; i++) {
    if (vutenti[i].l == 1) {
        utente_loggato = i;
    }
}

```

- verifica che l'orario corrente di prenotazione rientri negli orari di apertura del ristorante (dalle 7 alle 23)

```

var today = new Date();
var ora = today.getHours();
var fuoriOrario = false;
if (ora < 7 || ora > 23) {
    fuoriOrario = true;
}
if (fuoriOrario==true) {
    alert("Il tuo ordine è fuori orario, riprova domani!");
    return false;
}

```

- verifica che l'orario corrente di prenotazione sommato al tempo di elaborazione dell'ordine rientri negli orari di apertura del ristorante (dalle 7 alle 23), in particolare questo calcolo viene effettuato attribuendo a ogni piatto, sia dell'ordine corrente sia degli ordini in elaborazione dal ristorante, una media di 7 minuti necessari per la preparazione

```

var tempoElaborazione = 0;
//per gli ordini passati
for(j=0; j<vristoranti[idris].ordiniDaConfermare.length; j++) {
    var piattiOrdineDaConfermare=vristoranti[idris].ordiniDaC
    onfermare[j].piattiOrdine;
    for (k=0; k<piattiOrdineDaConfermare.length; k++) {
        tempoElaborazione+=7*(piattiOrdineDaConfermare[k][1]);
    }
}
//per l'ordine corrente
for (a=0; a<piattiOrdine.length; a++) {
    tempoElaborazione += 7*(piattiOrdine[a][1]);
}
var totaleMinuti = (ora*60);
totaleMinuti += minuti;
totaleMinuti += tempoElaborazione;
var calcoloOra = Math.floor(totaleMinuti/60);
if (calcoloOra < 7 || calcoloOra > 23) {
    fuoriOrario = true;
}
if (fuoriOrario == true) {
    alert("Il tuo ordine è fuori orario, riprova domani!");
}

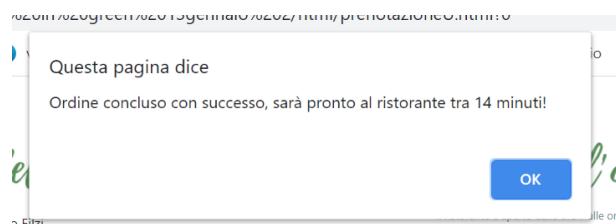
```

```

        return false;
    }

```

- verifica che il cliente abbia scelto un metodo di consegna, in caso contrario lo segnala con un alert
- inserisce i dati dell'ordine in due oggetti distinti
 - uno per il ristorante che lo memorizza nell'ultima posizione dell'array ordiniDaConfermare
 - uno per il cliente che lo memorizza nell'ultima posizione dell'array prenotazione
- segnala con un alert che l'ordine è stato confermato correttamente, indicando anche il tempo necessario per l'elaborazione del proprio ordine, se il metodo di consegna è il ritiro presso il ristorante, altrimenti l'utente viene indirizzato alla pagina mappa.html



ordiniU.html

In questa pagina il cliente visualizza la cronologia dei suoi ordini grazie alla funzione Javascript stampaPrenotazioniUtente() che, dopo aver individuato quale cliente è loggato, cicla le sue prenotazioni e ne stampa i dati.

Home Ordina ora Cronologia degli ordini

I tuoi ordini

Nome del ristorante: All in green di via Fabio Filzi
Data dell'ordine: 14:11 del 14/1/2021
Il tuo ordine sarà consegnato alle 14:25
Piatti: Pasta al pesto x2
Totale: 12 euro
Metodo di pagamento: pagamento alla consegna
Metodo di consegna: ritiro presso il ristorante

[Cancella l'ordine dalla tua cronologia](#) [Scrivi una recensione!](#)

[Nuovo ordine](#)

Il cliente può cancellare l'ordine dalla sua cronologia tramite il bottone Cancella l'ordine dalla tua cronologia che richiama la funzione Javascript cancellaPrenotazione(i, j), che prende in input la posizione del cliente all'interno del localStorage e la posizione della prenotazione all'interno dell'array prenotazione.

```
vutenti[i].prenotazione.splice(j,1);
```

Il cliente può scrivere una recensione tramite il bottone `Scrivi una recensione` che richiama la funzione Javascript `scriviRecensione(posizionePrenotazione)`, che prende in input la posizione della prenotazione all'interno dell'array `prenotazione` del cliente. Questa funzione indirizza il cliente alla pagina `recensioneU.html` inserendo nell'url `posizionePrenotazione` così che tramite questa si possa identificare il ristorante recensito tramite confronto tra email.

Il cliente può effettuare un nuovo ordine tramite il bottone `Nuovo ordine` che richiama la funzione Javascript `prenotaOra()`. Questa funzione porta il cliente alla pagina `scegliRistoranteU.html`.

mappa.html

Il cliente dopo aver effettuato l'ordine, se ha scelto come metodo di consegna "consegna a domicilio", viene rimandato alla pagina `mappa.html` in cui viene calcolato il percorso dal ristorante all'indirizzo del cliente (il percorso viene calcolato secondo le tempistiche di una bicicletta perché si tratta di un servizio green). Inoltre vengono effettuate delle modifiche:

- al tempo di attesa, sommando il tempo di preparazione dell'ordine con il tempo per percorrere la distanza tra i due indirizzi

```
var totale = Math.floor(data.duration / 60) +  
vutenti[utente_loggato].prenotazione[vutenti[utente_loggato].p  
renotazione.length - 1].tempoelaborazione;  
vutenti[utente_loggato].prenotazione[vutenti[utente_loggato].p  
renotazione.length - 1].tempoelaborazione = totale;
```

- al prezzo totale dell'ordine, sommando il totale dei piatti con il prezzo di consegna equivalente a 2 euro per chilometro percorso

```
var costiConsegna = Math.floor(data.distance/1000) * 2;  
vutenti[utente_loggato].prenotazione[vutenti[utente_loggato].p  
renotazione.length - 1].totale =  
vutenti[utente_loggato].prenotazione[vutenti[utente_loggato].p  
renotazione.length - 1].totale + costiConsegna;
```

Se la distanza da percorrere 50 chilometri, l'ordine non può essere effettuato e quindi viene eliminato dalla lista `ordiniDaConfermare` del ristorante e dalla lista `prenotazione` dell'utente

Per costruire la mappa e calcolare i tempi di percorrenza oltre alla distanza tra i due indirizzi, abbiamo utilizzato la piattaforma open source mapbox (<https://www.mapbox.com/>).

Per visualizzare la mappa con la posizione del ristorante prendiamo la variabile `address` stringa che contiene l'indirizzo del ristorante (preso dal `localStorage` dei ristoranti e individuato tramite l'invio della posizione del ristorante all'interno del `localStorage` dalla pagina `prenotazioneU.html`). Dopodiché trasformiamo questo indirizzo in coordinate geografiche tramite `geocodeForward` che salviamo nella variabile `coordinates`. A questo punto creiamo la mappa `map` e fissiamo il `center` con le coordinate del ristorante.

```
var client = new MapboxClient(mapboxgl.accessToken);
```

```

var address = vristoranti[numeroRist].indirizzo + " " +
vristoranti[numeroRist].citta + " " + vristoranti[numeroRist].cap;

client.geocodeForward(address, function(err, data, res) {
  var coordinates = data.features[0].center;
  var map = new mapboxgl.Map({
    container: "map",
    style: "mapbox://styles/mapbox/streets-v10",
    center: coordinates,
    zoom: 14
  });
  new mapboxgl.Marker()
    .setLngLat(coordinates)
    .addTo(map);
});

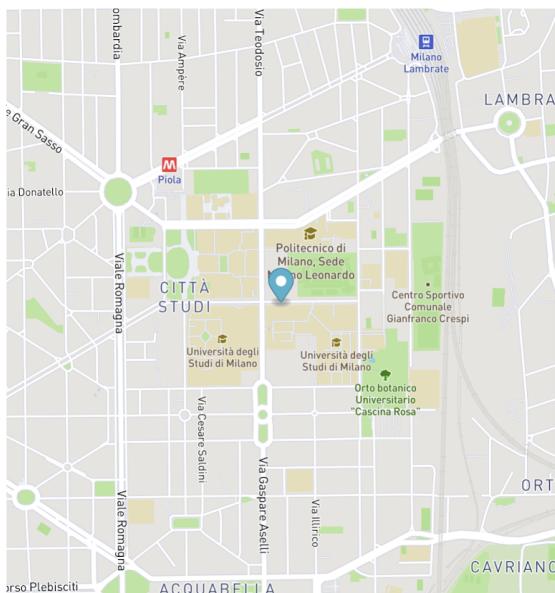
```



[Home](#) [Ordina ora](#) [Cronologia degli ordini](#)



*Ordine concluso
con successo!*



Per essere più 'green' l'ordine ti verrà recapitato in bicicletta 🚲

La preparazione del tuo ordine richiede: 133 min
La consegna del tuo ordine richiede: 63 min

Totale stima di consegna: 196 min
Totale costi di consegna: 24 €

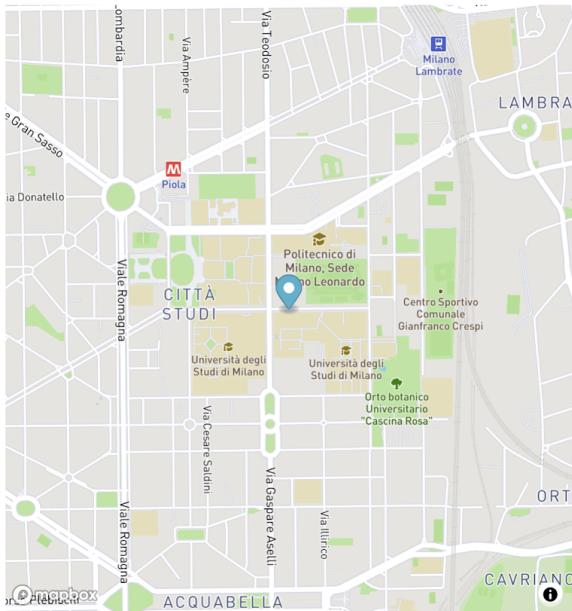
[Torna alla home](#)



Home Ordina ora Cronologia degli ordini



*L'ordine non è andato
a buon fine..*



Ci dispiace ma l'ordine non è stato effettuato perché l'indirizzo di consegna supera la distanza massima di 50 km.

[Torna alla home](#)

recensioneU.html

In questa pagina si trova un form in cui il cliente può inserire un titolo, una valutazione (tramite la selezione del numero di stelle) e un commento relativo al ristorante presso cui ha effettuato un ordine. Quando il cliente invia la sua recensione, la funzione Javascript `salvaRecensione()` inizialmente crea un oggetto con i seguenti campi:

```
var o = {  
    nome: nomeUtente,  
    cognome: cognomeUtente,  
    valutazione: valutazione,  
    titolo: document.recensione.titolo.value,  
    commento: document.recensione.commento.value,  
};
```

Successivamente individua la posizione nel `localStorage` del ristorante che recensito per poter salvare l'oggetto `o` all'interno dell'array `recensioni`. Per individuare la posizione la funzione fa un confronto tra l'email del ristorante salvata nell'ordine del cliente e le email di tutti i ristoranti del `localStorage`.

Titolo:

Valutazione:

Commento:

Invia **Cancella**

Annulla e torna indietro

viewRecensioniR.html

In questa pagina l'utente ristoratore può visualizzare le recensioni lasciate dai propri clienti ed ha la possibilità di eliminare queste se le ritiene non opportune da far visualizzare agli altri clienti.



Nome: MarioRossi

Valutazione: ★

Titolo: Cibo freddo!

Commento: La consegna del cibo è arrivata in ritardo e il cibo era freddo. Peccato sembravano piatti buoni

Cancella recensione dalla tua cronologia

Nome: AliceBianchi

Valutazione: ★★★★

Titolo: Buonissimo!

Commento: Ho ritirato l'ordine al ristorante e l'ho mangiato in pausa pranzo in un parco poco distante. Apprezzatissimo!

Cancella recensione dalla tua cronologia

Torna alla home

confermaOrdiniR.html

In questa pagina il ristoratore ha modo di visualizzare gli ordini che hanno effettuato i clienti e una volta conclusi può eliminarli da questa lista premendo il bottone "Ordine effettuato".



Home Recensioni Conferma gli ordini Statistiche vendite

Conferma gli ordini

Ordine effettuato il 14/1/2021, alle ore 15:49

Nome cliente: Mario Rossi

Telefono cliente: 333456789

Email cliente: mario.rossi@gmail.com

Indirizzo di consegna: via Celoria 18, Milano (20090)

Tipologia di consegna: consegna a domicilio

Ordine:

- Patate al forno x 1

Ordine effettuato

Ordine effettuato il 14/1/2021, alle ore 15:55

Nome cliente: Alice Bianchi

Telefono cliente: 333444555667

Email cliente: alice.bianchi@hotmail.com

Indirizzo di consegna: viale Venezia 12, Corsico (20345)

Tipologia di consegna: consegna a domicilio

Ordine:

- Patate al forno x 7

Ordine effettuato

statisticheR.html

Una volta che gli ordini sono stati conclusi, il ristoratore può visualizzare in questa pagina i grafici delle vendite avvenute settimanalmente e mensilmente nella propria attività avvenute nell'anno corrente. Abbiamo realizzato questi grafici mediante l'uso della libreria open source Chart JS (<https://www.chartjs.org/>).

I due grafici sono costruiti allo stesso modo ma comprendono dati differenti. Di seguito descriviamo il grafico delle vendite settimanali.

Tramite il tag “canvas” visualizziamo il grafico, in questo caso visualizziamo il grafico “myCanvas”.

```
<canvas id="myCanvas" ></canvas>
```

Dopodichè costruiamo il grafico all'interno di uno script. Per poter calcolare il numero delle vendite abbiamo creato una variabile per ogni giorno della settimana inizializzata a 0 in modo tale che per ogni data degli ordini presenti nella lista `ordiniEffettuati`, la variabile aumenti il suo conteggio di uno (tutto questo dopo aver verificato che la data dell'ordine sia nell'anno corrente).

```

var annoAttuale = new Date().getFullYear();
var countLun = 0;
var countMart = 0;
var countMerc = 0;
var countGiov = 0;
var countVen = 0;
var countSab = 0;
var countDom = 0;
for(i=0; i<vristoranti[ristorante_loggato].ordiniEffettuati.length;
i++) {
    var data = new Date(vristoranti[ristorante_loggato].ordiniEffettuati[i].anno + "/" +
    vristoranti[ristorante_loggato].ordiniEffettuati[i].mese + "/" +
    vristoranti[ristorante_loggato].ordiniEffettuati[i].giorno);

    if (data.getFullYear() == annoAttuale) {
        if(data.getDay() == 1) { //lunedì
            countLun += 1;
        } else if (data.getDay() == 2) {
            countMart += 1;
        } else if (data.getDay() == 3) {
            countMerc += 1;
        } else if (data.getDay() == 4) {
            countGiov += 1;
        } else if (data.getDay() == 5) {
            countVen += 1;
        } else if (data.getDay() == 6) {
            countSab += 1;
        } else if (data.getDay() == 0) {
            countDom += 1;
        }
    }
}
}

```

Definiti i conteggi delle vendite possiamo passare alla definizione del grafico definendo le labels e i dati che vogliamo rappresentare (ovvero i conteggi).

```

let myCanvas = document.getElementById("myCanvas").getContext('2d');
let myLabels = ["Lunedì", "Martedì", "Mercoledì", "Giovedì",
"Venerdì", "Sabato", "Domenica"];
let myData = [countLun, countMart, countMerc, countGiov, countVen,
countSab, countDom];

```

Di seguito vi è l'implementazione vera e propria del grafico con le seguenti attributi di visualizzazione. Noi abbiamo scelto un grafico di tipo lineare in cui si possa visualizzare, al passaggio del mouse sui valori, il valore delle vendite e il relativo giorno della settimana.

```
let chart = new Chart(myCanvas, {
    type: 'line', // tipo di grafico
    data: {
        labels: myLabels,
        datasets: [
            {
                label: "Numero di vendite giornaliere",
                backgroundColor: '#90C744',
                borderColor: '#90C744',
                fill: false,
                data: myData,
            }
        ],
        options: {
            title: { // titolo del grafico
                display: true,
                text: 'Grafico delle vendite settimanali dell\'anno ' +
                    annoAttuale,
                fontSize: 22,
                padding: 20
            },
            legend: {
                display: true, // per visualizzare legenda
                position: 'bottom' // di default è sopra
            },
            layout: {
                padding: 50 // compare il grafico appena lo vedi
            },
            tooltips: {
                enable: true
            },
            scales: {
                xAxes: [
                    {
                        display: true,
                    }],
                yAxes: [
                    {
                        display: true
                    }]
            },
            responsive: true,
            maintainAspectRatio: true
        }
    });
});
```



Home Recensioni Conferma gli ordini Statistiche vendite

Statistiche delle vendite

Controlla le statistiche della tua attività per migliorare il tuo servizio.

Grafico delle vendite settimanali dell'anno 2021

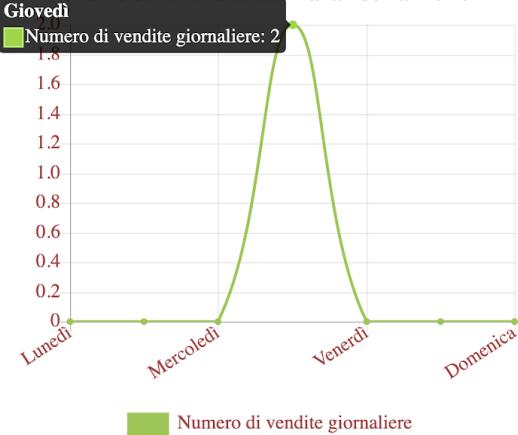


Grafico delle vendite mensili dell'anno 2021

