

Relazione per Fast Food

Click&Food

indice

- Introduzione	
1. Autore.....	pag 2
- Struttura dell'Applicazione	
1. Front-End.....	pag 2
2. Back-End.....	pag 3
3. NODEJS.....	pag 3
- NODEJS ed EXPRESS.....	pag 5
- File NODEJS.....	pag 6
- Database MONGODB.....	pag 7
- Gestione del profilo dell' utente.....	pag 7
- Gestione del Ristorante.....	pag 9
- Gestione degli Ordini.....	pag 11
- consegna a asporto.....	pag 13
- aggiunta e cancellazione piatti.....	pag 14
- modifica profilo user.....	pag 16
- Configurazione dell'applicazione.....	pag 16
- Scelte implementative e features.....	pag 17
- Swagger JS.....	pag 17
- File SWAGGER.JS.....	pag 18
- Interfaccia grafica SWAGGER.....	pag 19
- Installazione.....	pag 19
- Gestione codici HTTP.....	pag 21
- Menù.....	pag 22
- Lingua.....	pag 23

introduzione:

Questo documento rappresenta la relazione del progetto "Fast food" che ho deciso di chiamare Click&Food, sviluppato nel contesto del corso "Programmazione Web e mobile" durante l'anno accademico 2024/2025 .

- progetto è stato realizzato da: Thomas Baiocchi (Matricola: 0158A)
-

Struttura dell'Applicazione

Front-End:

front-end di una pagina web è la parte dell'applicazione con cui gli utenti interagiscono direttamente. Include il design, la struttura, i colori, i font, i pulsanti e qualsiasi altro elemento visivo e interattivo.

All'interno della directory, sono presenti i seguenti elementi principali:

- **/frontend/**: Questa directory contiene i file HTML che vengono renderizzati da browser, determinando quindi interfaccia grafica dell'applicazione. Tutti i files sono stati validati per lo standard HTML5.
- **/css/**: Questa directory contiene i file di stile che definiscono l'aspetto visivo dell'applicazione. Tutti i files sono stati validati per lo standard CSS3. I file principali sono:

/Frontend/css/style.css - Questo file definisce lo stile generale dell'applicazione

- **/scripts/**: Questa directory contiene file JavaScript che gestiscono la logica del Front-End. Alcuni dei file principali includono:

/frontend/js/add.js - Permette a un ristoratore di aggiungere piatti al proprio menù, dal menù comune oppure aggiungendo un piatto che non fa parte della lista comune

/frontend/js/edit.js - Permette le modifiche su un piatto esistente del ristorante,

/frontend/js/login.js - Questo file si occupa di tutte le operazioni necessarie al login dell'utente

/frontend/js/main.js - Gestisce tutta la parte di visualizzazione dei menù e offerte speciali che funziona quando utente che specifica la preferenza di piatto mostra tutti i ristoranti che fanno piatti con quell'ingrediente tipo(carne,pollo,vegano oppure vegetariano), ed a delle funzionalità diverse per cliente, ristoratore e visitatore.

/frontend/js/order.js -Mostra i piatti raggruppati per ristorante, permette di ordinare i piatti e calcola il totale in tempo reale e il pagamento (solo utenti loggati).

/frontend/js/ordini_ristoranti.js -Mostra al ristoratore gli ordini che includono i suoi piatti e permette di avanzarne lo stato.

/frontend/js/payment.js -Valida i dati della carta, completa il pagamento e crea l'ordine sul backend.

/frontend/js/register.js -Gestisce la registrazione di clienti e ristoratori con campi dinamici differenti per i clienti vengono richiesti il nome il cognome il metodo di pagamento e le preferenze di validazione base.

/frontend/js/restaurant.js -gestisce le operazioni CRUD (Create, Read, Update, Delete) sui ristoranti, salvandoli in un file JSON locale

/frontend/js/ricerca_piatti.js -lo script che popola la pagina “Lista Piatti” con i piatti del ristorante del ristoratore e permette di modificarli o cancellarli.

/frontend/js/ricerca_ristoranti.js - Gestisce la ricerca dinamica dei ristoranti e aggiorna la pagina con i risultati che sono filtrati in base alla città alla categoria di alimenti che cucina .

/frontend/js/statistiche_ristoratore.js - Questo script permette al ristoratore di visualizzare report e statistiche di vendita sul proprio ristorante.

/frontend/js/storico_piatti.js - Questo script permette al cliente di vedere sia gli ordini in corso sia lo storico degli ordini completati.

La suddivisione chiara tra file HTML, file CSS e file JavaScript consente una gestione efficiente del Front-End e garantisce un'esperienza utente di alta qualità.

Back-End

back-end di una pagina web è la parte dell'applicazione che gestisce la logica di business, l'elaborazione dei dati e la comunicazione con il database. È responsabile di ricevere richieste dal front-end, elaborarle e restituire le risposte appropriate.

Componenti principali del back-end:

- **Server** → Gestisce le richieste HTTP e instrada i dati
 - **Database** → Memorizza e gestisce i dati
 - **API** → Intermediario tra front-end e database
 - **Autenticazione e Sicurezza** → Gestisce login, autorizzazioni e protezione dei dati
-

NODEJS

COS'E' NODEJS ED EXPRESS

Node.js ed Express costituiscono un binomio potente nell'ambito dello sviluppo web di applicazioni scalabili ed efficienti.

- Node.js è un runtime JavaScript asincrono e basato su eventi, costruito sul motore V8 di Chrome, che permette di eseguire codice JavaScript lato server.
caratteristiche: (Event-driven e Non-Blocking I/O; Single-threaded con Event Loop; Node Package Manager; utilizzato per API, Web Server e Microservizi)
- Express è un framework web minimalista e veloce per Node.js, utilizzato per costruire API e applicazioni web lato server.
caratteristiche: (leggero e minimalista; routing flessibile; middleware support; compatibilità con template engine; facile integrazione con database).

File NODEJS

swagger.js

swagger_output.js

/backend /: Questa cartella contiene i file dedicati alla configurazione dell'applicazione, ad eccezione delle variabili d'ambiente. Al suo interno, sono presenti:

Inoltre Alcuni dei file e delle directory principali sono:

/backend/login.js : Si occupa di ricevere username e password dal corpo della richiesta e legge tutti gli utenti dal file poi cerca se esiste un utente con stessa username e password se non lo trova → risponde con errore 401 Unauthorized invece se lo trova → restituisce in JSON solo username e role dell'utente.

`/backend/meals.js`: In pratica meals è il “cuore” del backend dei piatti: gestisce CRUD completo (crea, legge, aggiorna, elimina) e funziona sia con database MongoDB che con file locale, così il progetto rimane operativo anche senza DB.

`/backend/db.js`: in pratica è il punto di ingresso: accende il server, connette il DB, espone le API, fornisce Swagger e gestisce shutdown ordinato.

`/backend/orders.js`: In pratica crea un nuovo ordine; prende i piatti dal carrello, li arricchisce con nome/prezzo, calcola il totale, deduce userId e restaurantId se mancano, normalizza consegna/pagamento e salva su MongoDB ed elenca ordini: con filtri opzionali aggiorna stato: controlla che la transizione sia valida, aggiorna lo stato, aggiunge timestamp, ha helper per leggere utenti/piatti dai file JSON come fallback, unificare carrelli in vari formati, normalizzare dati e validare l’ordine.

`/backend/register.js`: in poche parole gestisce la registrazione utenti, controlla che username, email e password ci siano e che non siano già usati, se l’utente è un ristoratore, assegna un restaurantId e salva anche i dati del ristorante (nome, telefono, partita IVA, indirizzo, luogo), se invece è un cliente, può avere una preferenza di categoria (preferenza).

`/backend/restaurant.js`: in pratica dei ristoranti, get /restaurant → restituisce la lista, con possibilità di filtro per nome o luogo, get /restaurant/:id → restituisce un singolo ristorante cercandolo tramite restaurantId, post /restaurant → crea un nuovo ristorante, put /restaurant/:id → aggiorna i dati di un ristorante esistente tramite restaurantI

`/backend/users.js`: gestisce registrazione, login, modifica, cancellazione utenti e lista ristoratori.

`index.js`: Questo file rappresenta il punto di ingresso principale dell'applicazione, contenente le istruzioni per l'avvio dell'app e la definizione degli endpoint. La struttura ben organizzata del Back-End garantisce una gestione efficiente delle funzionalità server-side e contribuisce al corretto funzionamento dell'applicazione.

Database MONGODB

Nel corso di sviluppo dell'applicazione, è stato fatto largo uso del database MongoDB. Qui di seguito, vengono presentate le collezioni sono state create e utilizzate per immagazzinare i dati essenziali dell'applicazione.

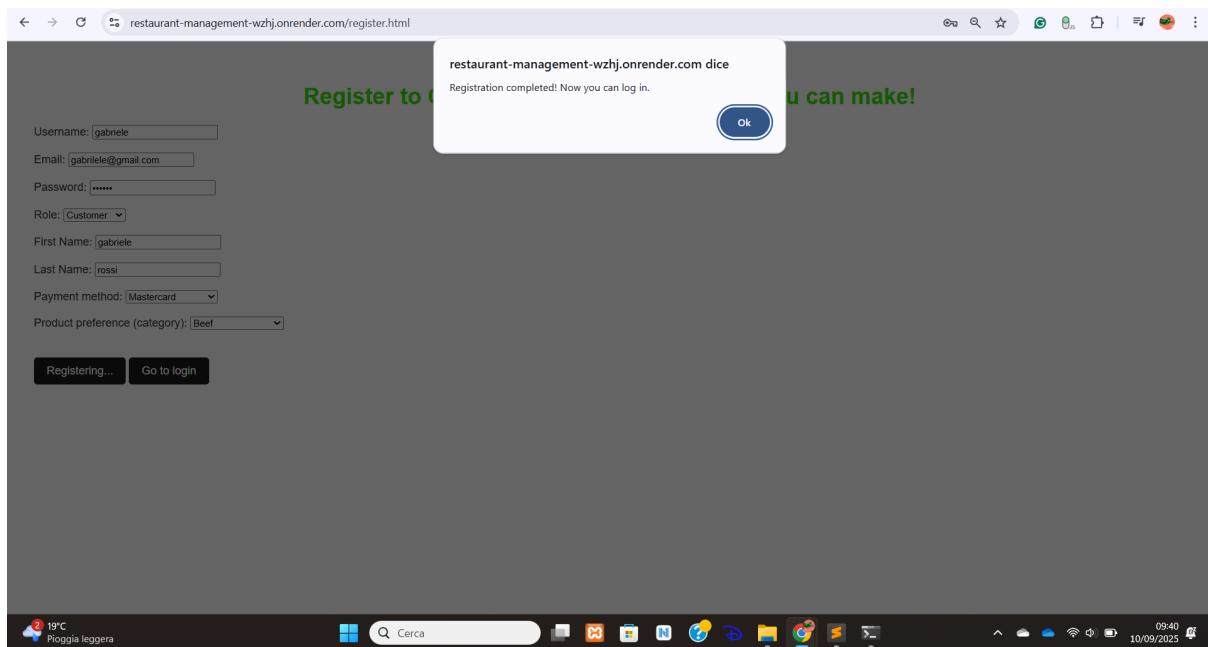
MongoDB è un database NoSQL orientato ai documenti, progettato per gestire grandi volumi di dati in modo scalabile e flessibile. Caratteristiche:(Basato su documenti; Schema flessibile; scalabilità orizzontale;alta velocità e prestazioni; query avanzate e indicizzazione)

Per questa applicazione sono state utilizzate le seguenti collections:

- albums: Collezione che gestisce informazioni degli album di figurine
- cards: Collezione che gestisce le carte presenti all'interno di ogni album
- users: Collezione che gestisce i dati degli utenti
- exchanges: Collezione che gestisce gli scambi proposti dagli utenti
- exchanges_cards: Collezione che gestisce le carte proposte all'interno degli scambi.

Di seguito viene riportata una descrizione delle collections e del loro schema.

Gestione del profilo dell' utente: parte dalla registrazione del utente al sito dove verranno chiesti un username ed una password ed un indirizzo email come informazioni in più se utente è un customer chiede il nome il cognome il metodo preferito per il pagamento ed inoltre chiede anche la categoria alimentare preferita come (carne,pesce,pollo, ecc) ecco un esempio se utente è un cliente



se la registrazione va buon fine rimanda il customer alla login dove li potra accedere ai menù ed ordinare i piatti che desidera utente.

Gestione del Ristorante: parte dalla registrazione del utente al sito dove verranno chiesti un username ed una password ed un indirizzo email come informazioni in più se utente è un ristorante chiede il nome del ristorante la partita IVA il numero di telefono la città in cui è il ristorante ed la via a cui appartiene
ecco un esempio se utente è un ristorante

← → ⌛ restaurant-management-wzhj.onrender.com/register.html

Register to Click&Food — the best choice you can make!

Username:

Email:

Password:

Role:

Restaurant name:

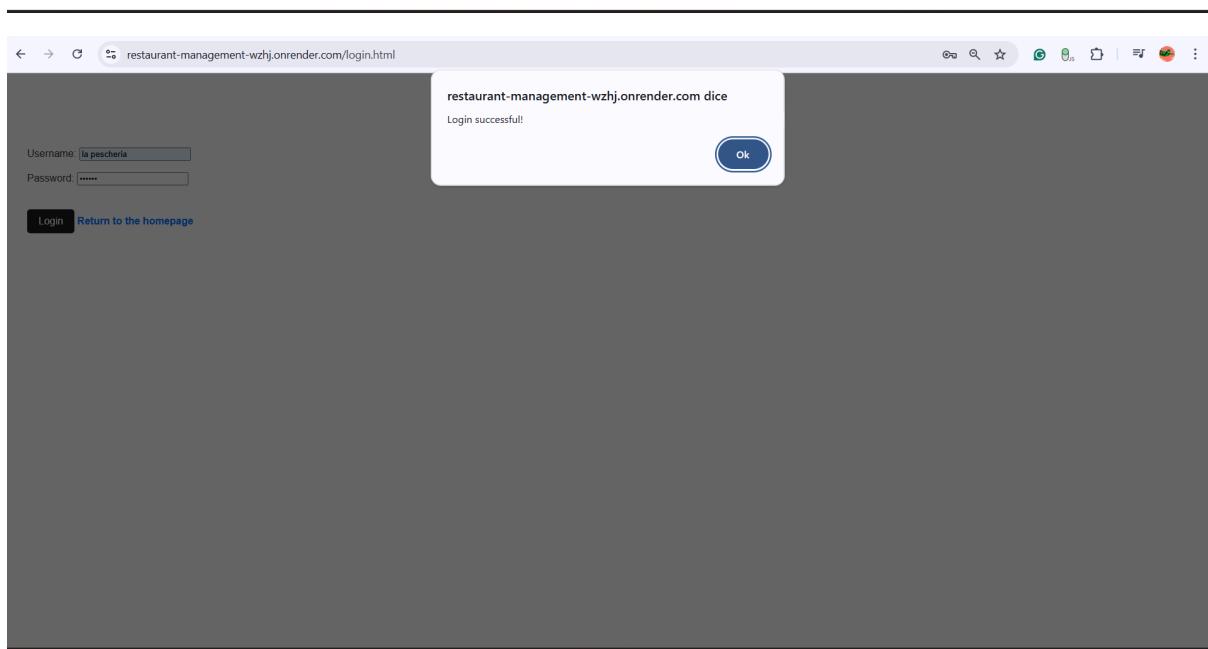
VAT number:

Phone number:

City:

Street address:

[Register](#) [Go to login](#)



ecco come si presenta la homepage se utente è un ristoratore

restaurant-management-wzhj.onrender.com

Welcome to Click&Food

All the best restaurants just a click away!!!!

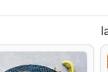
Welcome, la pescheria

Add Dish | Edit Profile | Logout

Search for dishes Orders received Statistics

Restaurant menu

la pescheria Ristorante

 <p>Cajun spiced fish tacos Seafood</p> <p>Ingredients: Cajun, Cayenne pepper, white fish, vegetable oil, flour tortilla, avocado, little gem lettuce, Spring Onion, salsa, sour cream, lemon, garlic.</p> <p>€ 10.00</p>	 <p>Baked salmon with fennel & tomatoes Seafood</p> <p>Ingredients: Fennel, Parsley, Lemon, Cherry Tomatoes, Olive Oil, Salmon, Black Olives</p> <p>€ 10.00</p>	 <p>Nasi lemak Seafood</p> <p>Ingredients: Coconut Milk, Water, Ginger Paste, Ginger, Bay Leaf, Rice, Eggs, Cucumber, Peanuts, Anchovy, Lemongrass, Vegetable Oil, Onion, Garlic, Shallots, Chili Powder, Anchovy Paste, Sugar, Tamarind Paste</p> <p>€ 10.00</p>	 <p>Grilled Portuguese sardines Seafood</p> <p>Ingredients: Sardines, Olive Oil, Garlic, Paprika, Lemon, Rosemary, Red Chili Flakes</p> <p>€ 10.00</p>	 <p>Fish fofos Seafood</p> <p>Ingredients: Haddock, Potatoes, Green Chilli, Coriander, Cumin Seeds, Pepper, Garlic, Ginger, Flour, Eggs, Breadcrumbs, Vegetable Oil</p> <p>€ 10.00</p>
---	---	---	--	--

Gestione degli ordini: se un customer vuole ordinare i piatti che vengono raggruppati per ristoranti per il customer basta schiacciare order sotto i piatti che vengono aggiunti in order che poi permette di aggiungere altri piatti schiacciando add dish,
ecco un esempio se utente è un cliente che può ordinare

dagabri Ristorante

			
Apam balik	Bean & Sausage Hotpot	Boaty Breakfast	Big Mac
Ingredients: Milk, Oil, Eggs, Flaking Powder, Salt, Unsalted Butter, Sugar, Peanut Butter € 10.00	Ingredients: Sausage, Beans, Black Truffle, Egg, Tomato, Mustard € 10.00	Ingredients: Popovers, Spring Onions, Plain Flour, Egg White, Milk, Bicarbonate of Soda, Chopped Tomatoes, Dill, Cherry Tomatoes, Bacon, Egg € 10.00	Ingredients: Minced Beef, Olive Oil, Sesame Seed Burger Buns, Onion, Iceberg Lettuce, Cheese, Dill Pickles, Mayonnaise, White Wine Vinegar, Pepper, Mustard, Onion Salt, Garlic Powder, Paprika € 10.00
Order	Order	Order	Order

La crème Ristorante

		
Apple Frangipane Tart	Apam balik	Banana Pancakes
Ingredients: digestive biscuits, butter, eggs, apples, salted butter, caster sugar, flour, ground almonds, custard powder, hazelnut almonds € 10.00	Ingredients: Milk, Oil, Eggs, Flaking Powder, Salt, Unsalted Butter, Sugar, Peanut Butter € 10.00	Ingredients: Banana, Eggs, Milk, Oil, Cinnamon Extract, Oil, Pecan Nuts, Raspberries € 10.00
Order	Order	Order

Order

	Big Mac
Beef	Ingredients: Minced Beef, Olive Oil, Sesame Seed Burger Buns, Onion, Iceberg Lettuce, Cheese, Dill Pickles, Mayonnaise, White Wine Vinegar, Pepper, Mustard, Onion Salt, Garlic Powder, Paprika
Unit price	€10.00
Quantity <input type="button" value="1"/>	

Add dishes

Delivery method

Pick up at the restaurant

Payment method

Credit card
 Prepaid card

Order summary

You have an in-progress order for another restaurant. It won't be included.

Big Mac x 1	€10.00
Subtotal	€10.00
Fees	€0.00
Total	€10.00

[Add to the order](#) [Confirm order](#)

Ecco un esempio di pagamento

Payment by card

Card number:

Card holder:

Expiry:

CVV:

[Pay now!!](#)

Invece ecco la finestra dove da la conferma del ordine ed indica anche il tempo quando il piatto è pronto quindi alla scadenza è pronto

Order Confirmed

Thank you for choosing us!

Order ID: 9
 Date: 10/09/2025, 09:57:20
 Pickup: at the restaurant
 Payment method: [object Object]
 Estimated time: ~32 min (ready by 10:29)

Ordered dishes:

- Cajun spiced fish tacos × 2
- Grilled Portuguese sardines × 2

[Return to the homepage](#)

invece lato ristorante la pagina di riferimento ad ordini ricevuti dove elenca i piatti ed il nome del customer che sono stai ordinati dai clienti, e dove possiamo di avanzare lo stato dei piatti ordinati dai clienti.

The screenshot shows a web browser window with the URL restaurant-management-wzhj.onrender.com/ordini_ristoratore.html. The page title is "Orders received". The order details are as follows:

- Customer:** gabriele
- Status:** ordinato
- Delivery:** asporto
- Payment:** [object Object]
- Dishes:**
 - Cajun spiced fish tacos
 - Grilled Portuguese sardines

A black button labeled "Advance status" is visible at the bottom.

[Return to the homepage](#)

consegna a asporto: invece la consegna di asporto a come “cuore” my order dove il client può vedere gli ordini e segnalare quando ordine. ecco un esempio se utente è un cliente

The screenshot shows a web browser window with the URL restaurant-management-wzhj.onrender.com/i-miei-ordini.html. The page title is "My orders". The order details are as follows:

- Total spent (all orders): €20.00
- Active total: €20.00 — 1 active order
- ID: 9
- Created: 10/09/2025, 09:57:20
- Mode: Pick up in store
- Status: Ordered
- Estimated time: Ready for pickup
- Dishes:
 - Cajun spiced fish tacos ×1 — each €10.00 → €10.00
 - Grilled Portuguese sardines ×1 — each €10.00 → €10.00
- Order total: €20.00

A green button labeled "I picked up the order" is visible at the bottom.

[Return to the homepage](#)

[Order history](#)

quando si preme il bottone “i pick up the order” utente conferma di aver preso ordine ordine passa negli “order history”

The screenshot shows a web browser window with the URL restaurant-management-wzhj.onrender.com/i-miei-ordini.html. A confirmation dialog box is displayed in the center:

restaurant-management-wzhj.onrender.com dice
Are you sure you picked up your order at the counter?

Ok Annulla

The main page content includes:

- Total spent (all orders): €20.00
- Active total: €20.00 — 1 active order
- ID: 9
- Created: 10/09/2025, 09:57:20
- Mode: Pick up in store
- Status: Ordered
- Estimated time: Ready for pickup
- Dishes:
 - Cajun spiced fish tacos ×1 — each €10.00 → €10.00
 - Grilled Portuguese sardines ×1 — each €10.00 → €10.00
- Order total: €20.00

A green button labeled "I picked up the order" is visible at the bottom.

← → ⌛ restaurant-management-wzhj.onrender.com/storico_clienti.html

Order history

Active orders

- No orders —

Orders completed

- ID: 9
 Created: 10/09/2025, 09:57:20 · Closed: 10/09/2025, 16:42:10
 Status: Picked up
 Dishes:
 - Cajun spiced fish tacos ×1 — unit €10.00 → €10.00
 - Grilled Portuguese sardines ×1 — unit €10.00 → €10.00
 Total: €20.00
 Pickup at restaurant

[Return to my orders](#)

[Return to the homepage](#)

aggiunta e cancellazione piatti: per il ristorante ha la possibilità di aggiungere i piatti e togliere piatti con l'add possiamo aggiungere piatti schiacciando add in the menu il piatto viene aggiunto nel menù possiamo avere una alternativa aggiunta “manuale”.

← → ⌛ restaurant-management-wzhj.onrender.com/add.html

Add Dish

Name:

Price:

Description:

Category:

Image URL:

Ingredients:

ed invece per la cancellazione usiamo ricerca piatti schiacciando il pulsante “delete” dopo la conferma il piatto viene eliminato dal menù del ristorante

Name:

Type/size:

Maximum price:

Search for dishes

Perfect for you

[Search](#)

[Back to homepage](#)

[Results](#)

Restaurant
Mango - 23790556



Ann's Balik
Type: Dessert
Price: €10-20

Ingredients: Milk, Oil, Eggs, Flour, Baking Powder, Salt, Unsalted Butter, Sugar, Peppermint Butter

In a large casserole, mix the flour, baking powder, salt, sugar and butter. Mix until all ingredients are combined evenly. Spread some batter onto the pan. Spread a thin layer of batter to the side of the pan. Cover the pan for 30-60 seconds until small air bubbles appear. Add butter, cream corn, crushed peanuts and sugar onto the pancake. Fold the pancake into half once the bottom surface is browned. Cut into wedges and best eaten when it is warm.



Bean & Sausage Hotpot
Type: Main Course
Price: €10-20

Ingredients: Beans, Sausages, Tomato Sauce, Butter Beans, Black Treacle, English Mustard

In a large casserole, fry the sausages until brown all over – about 10 mins. Add the onion, saute, stirring well, then stir in the beans, treacle or sugar and mustard. Bring to the simmer, cover and cook for 30 mins. Great served with crusty bread or rice.



Boozy Breakfast
Type: Main Course
Price: €10-20

Ingredients: Bacon, Spring Onions, Plain Flour, Egg White, Milk, Bicarbonate of Soda, Butter, Vegetable Oil, Cherry Tomatoes, Bacon, Egg

STEP 1 Before you start, put your oven on its lowest setting, ready to keep things warm. Peel the potatoes, grate 2 of them, then set aside. Cut the other 2 into bite-sized, then boil for 10-15 mins or until tender. Meanwhile, squeeze a small amount of the liquid from the grated potatoes as you can using a sieve or hand towel. Mix the boiled potatoes, then mix with the grated potato, spring onions and flour. Add the bicarbonate of soda and mix well. Heat a griddle pan over a medium heat, then add a knob of butter and spread it around. Once the griddle is hot, add a few cherry tomatoes to the pan and fry until golden and crispy. Remove from the oven while you cook the next batch, adding more butter and oil to the pan before you do so. You will get 15 dumpling-size bites from the mix. Can be made the day ahead, stored on kitchen paper, then reheated in a low oven for 20 mins. STEP 4 Heat a griddle to medium and put the tomatoes in a heavy-based pan. Add a good knob of butter and a little oil, then fry for about 5 mins until charred and crisp. Serve with a fried egg and bacon.



Big Mac
Type: Fast Food
Price: €10-20

Ingredients: Minced Beef, Onions, Sesame Seed Burger Buns, Onion, Lettuce, Cheese, Dr. Pepper, Mayonnaise, White Wine Vinegar, Pepper, Mustard, Onion Salt, Garlic Powder, Paprika

1. Preheat the oven to 200°C. In a large bowl, combine all the ingredients in a bowl. Season with salt and add 1 egg to bind. Form into 4 balls using about 1/3 cup mixture each. Place onto a square of baking paper and flatten to form into four x 15cm circles. Heat oil in a large frying over high heat. In 2 batches, cook beef patties for 1-2 minutes each until lightly charred and cooked through. Remove from heat and keep warm. Repeat with remaining two patties. 3. Gently slice each burger bun into three ensaymadas, then lightly toast. 4. To assemble the burgers, spread a little Big Mac sauce over the bottom base. Top with some chopped onion, shredded lettuce, slice of cheese, beef patty and some potato slices. Top with the middle bun layer, and spread with more Big Mac sauce, onion, lettuce, potato, beef patty and then finish with more sauce. Top with burger lid to serve. 5. After warming half an hour for your food to settle, go for a gig.

La crema Restaurant
Mangia - 23790556

Apple Frangipane Tart
Type: Dessert
Price: €10-20



Ann's Balik
Type: Dessert
Price: €10-20

Ingredients: Double Cream, Butter, Baking apples, Salted Butter, Caster sugar, free-range eggs, teacan, ground almonds, almond extract, flaked almonds

Preheat the oven to 180°C. Line a 23cm tart tin with the shortcrust pastry. Beat the cream until soft peaks form. Add the double cream, butter and salted butter. Tip the tart tin in and, using back of the spoon, press over the base of the tin to give an even layer. Chill in the fridge while you make the filling. Cream together the butter and sugar until light and fluffy. You can do this in a food processor if you have one. Process for 2-3 minutes. Melt the eggs in the microwave for 10 seconds. Add the eggs to the cream and almond extract and blend until well combined. Peel the apples, and cut them into slices of apple. Do this at the last minute to prevent the apple going brown. Arrange the slices over the biscuit base. Spread the frangipane filling over the top of the tart. Scatter the flaked almonds over the tart. Sprinkle with a few more cream and sugar swirls with the flaked almonds. Bake for 20-25 minutes until golden-brown and set. Remove from the oven and leave to cool for 15 minutes. Remove the sides of the tin. An easy way to do this is to stand the tin on a can of beans and push down gently on the edges of the tin. Transfer the tart, with the base retained, to a serving plate. Serve warm with cream, fresh or ice cream.



Bean & Sausage Hotpot
Type: Main Course
Price: €10-20

Ingredients: Milk, Oil, Eggs, Butter, Baking Powder, Salt, Unsalted Butter, Sugar, Peanut Butter

Mix milk, oil and eggs together. Stir flour, baking powder and salt and the mixture. Stir until all ingredients are combined evenly. Spread some batter onto the pan. Spread a thin layer of batter to the side of the pan. Cover the pan for 30-60 seconds until small air bubbles appear. Add butter, cream corn, crushed peanuts and sugar onto the pancake. Fold the pancake into half once the bottom surface is browned. Cut into wedges and best eaten when it is warm.

restaurant-management-wzhj.onrender.com dice
Delete dish "Cajun spiced fish tacos"? This action cannot be undone.

Ok **Annulla**

Name:
Typology: seafood
Maximum price:

Search

[Back to homepage](#)

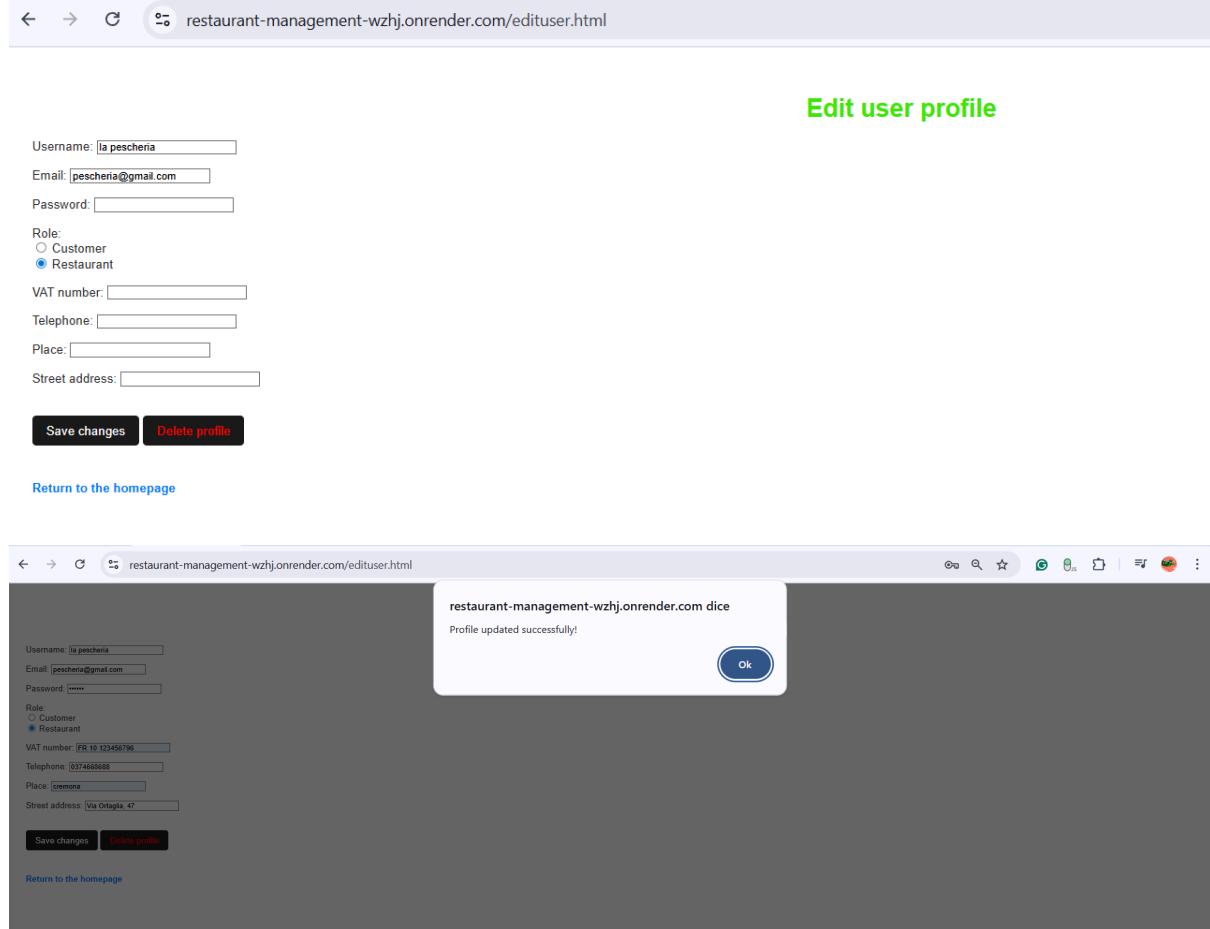
Results

la pescheria Ristorante
milano • 0374658673

 Cajun spiced fish tacos
Type: Seafood
Price: €
Seasonal: cajun, cayenne pepper, white fish, vegetable oil, flour tortilla, little gem lettuce, Spring Onions, salsa, sour cream, lemon, garlic
Cooking in a cajun spice and cayenne pepper marinade makes this fish super succulent and flavoursome. Top with a zesty dressing and serve in a tortilla for a quick, fuss-free main that's delightfully summery. On a large plate, mix the cajun spice and cayenne pepper with a little seasoning and use to coat the fish all over. Heat a little oil in a frying pan, add in the fish and cook over a medium heat until golden. Reduce the heat and continue frying until the fish is cooked through, about 10 minutes. Cook in batches if you don't have enough room in the pan. Meanwhile, prepare the dressing by combining all the ingredients with a little seasoning. Soften the tortillas by heating in the microwave for 5-10 seconds. Pile high with the avocado, lettuce and spring onion, add a spoonful of salsa, top with large flakes of fish and drizzle over the dressing.

[Delete](#)

Modifica profilo user: abbiamo anche la possibilità di modificare account sia se siamo dei client che siamo dei ristoranti ad ogni modifica esce avviso di correttezza di modifica del profilo



restaurant-management-wzhj.onrender.com/edituser.html

Edit user profile

Username:

Email:

Password:

Role:
 Customer
 Restaurant

VAT number:

Telephone:

Place:

Street address:

Save changes **Delete profile**

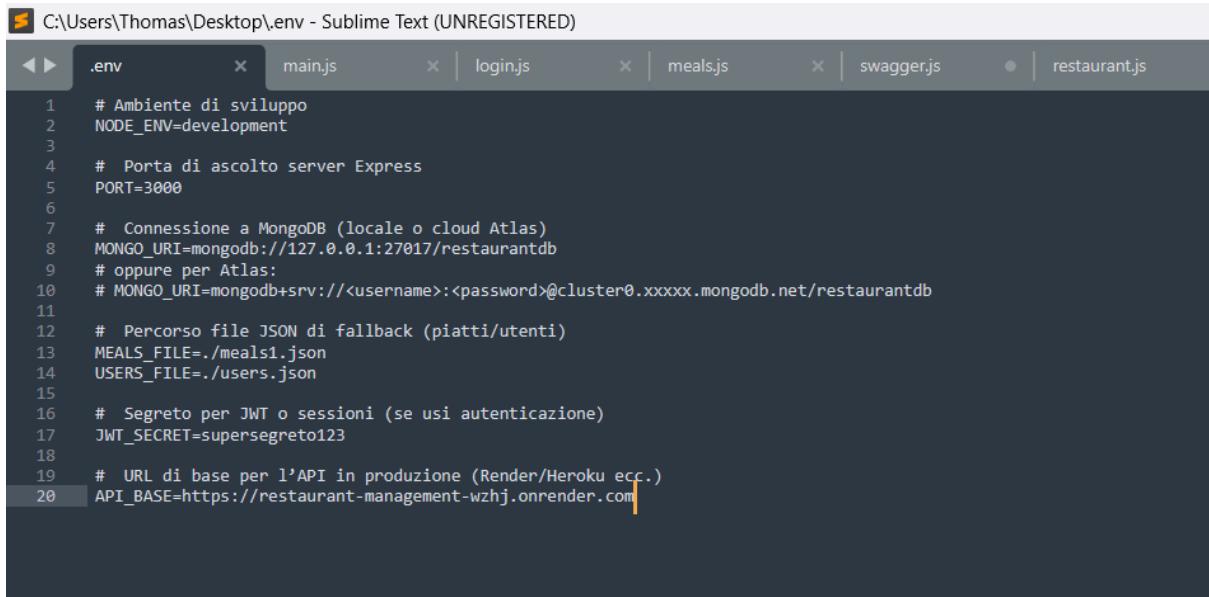
[Return to the homepage](#)

restaurant-management-wzhj.onrender.com dice
Profile updated successfully!

Ok

Configurazione dell'applicazione: Il progetto necessita di un file .env nella directory principale del dove sono contenuti i parametri necessari per il funzionamento.

Il file .env è gestito attraverso il pacchetto npm dotenv che si occupa di popolare le relative variabili d'ambiente e renderne semplice l'utilizzo e accesso tramite JavaScript.



```

C:\Users\Thomas\Desktop\.env - Sublime Text (UNREGISTERED)
.env      x  main.js      x  login.js      x  meals.js      x  swaggerjs      ●  restaurant.js

1 # Ambiente di sviluppo
2 NODE_ENV=development
3
4 # Porta di ascolto server Express
5 PORT=3000
6
7 # Connessione a MongoDB (locale o cloud Atlas)
8 MONGO_URI=mongodb://127.0.0.1:27017/restaurantdb
9 # oppure per Atlas:
10 # MONGO_URI=mongodb+srv://<username>:<password>@cluster0.xxxxx.mongodb.net/restaurantdb
11
12 # Percorso file JSON di fallback (piatti/utenti)
13 MEALS_FILE=./meals1.json
14 USERS_FILE=./users.json
15
16 # Segreto per JWT o sessioni (se usi autenticazione)
17 JWT_SECRET=supersegreto123
18
19 # URL di base per l'API in produzione (Render/Heroku ecc.)
20 API_BASE=https://restaurant-management-wzhj.onrender.com

```

Scelte implementative e features

Swagger JS: è un framework open-source per la progettazione, la creazione e la documentazione di API RESTful.

La sua utilità si concentra sulla semplificazione del processo di sviluppo API, consentendo agli sviluppatori di definire chiaramente le specifiche delle API, testarle e generare automaticamente documentazione dettagliata.

Per la generazione dello swagger ho utilizzato il module swagger-autogen. Tramite la creazione di un file swagger.js con una apposita configurazione e determinati commenti nella sezione degli endpoint, è possibile generare automaticamente una documentazione per gli endpoint.

è possibile visualizzare lo swagger generato all'endpoint /api-docs

File SWAGGER.JS

NB: Il codice riportato di seguito rappresenta un esempio molto vicino a quello utilizzato in questa applicazione!

```

◀ ▶ .env      x | main.js      x | login.js      x | meals.js      x | swagger.js   ● | restaurant.js  x | users.js      x | index.html  x | ordini_ristorato
1  openapi: 3.0.0
2  info:
3    title: Click&Food API
4    version: 1.0.0
5    description: >
6      API per la gestione di ristoranti, piatti e utenti nel progetto **Click&Food**.
7      Supporta funzioni CRUD per i piatti e la gestione di utenti con ruoli diversi (cliente / ristoratore).
8    servers:
9      - url: http://localhost:3000
10     description: Server locale
11     - url: https://restaurant-management-wzhj.onrender.com
12     description: Server in produzione
13
14   tags:
15     - name: Users
16       description: Gestione utenti (registrazione, login)
17     - name: Meals
18       description: Gestione piatti (CRUD)
19
20   paths:
21     /users:
22       get:
23         tags: [Users]
24         summary: Restituisce la lista degli utenti registrati
25         responses:
26           '200':
27             description: Lista utenti
28       post:
29         tags: [Users]
30         summary: Registra un nuovo utente
31         requestBody:
32           required: true
33           content:
34             application/json:
35               schema:
36                 $ref: '#/components/schemas/User'
37         responses:
38           '201':
39             description: Utente creato
40           '400':
41             description: Dati non validi
42
43     /users/login:
44       post:
45         tags: [Users]
46         summary: Login utente
47         requestBody:
48           required: true
49           content:
50             application/json:
51               schema:
52                 type: object
53                 properties:
54                   username:
55                     type: string
56                   password:
57                     type: string
58         responses:
59           '200':
60             description: Login riuscito
61           '401':
62             description: Credenziali non valide
63
64     /meals:
65       get:
66         tags: [Meals]
67         summary: Restituisce tutti i ristoranti con i loro piatti
68         responses:
69           '200':
70             description: Lista ristoranti con menu
71       post:
72         tags: [Meals]
73         summary: Aggiunge un nuovo piatto a un ristorante
74         requestBody:
75           required: true
76           content:
77             application/json:
78               schema:
79                 $ref: '#/components/schemas/Meal'
80         responses:
81           '201':
82             description: Piatto creato
83           '400':
84             description: restaurantId mancante
85           '404':
86             description: Ristorante non trovato
87
88     /meals/{id}:
89       get:
90         tags: [Meals]
91         summary: Restituisce un piatto specifico
92         parameters:
93           - in: path
94             name: id
95             required: true
96             schema: { type: integer }
97         responses:
98           '200':
99             description: Piatto trovato
100           '404':
101             description: Piatto non trovato
102       put:
103         tags: [Meals]
104         summary: Aggiorna un piatto esistente
105         parameters:
106           - in: path
107             name: id
108             required: true
109             schema: { type: integer }
110         requestBody:

```

```
110     schema: { type: Integer }
111   required: true
112   content:
113     application/json:
114       schema:
115         $ref: '#/components/schemas/Meal'
116   responses:
117     '200':
118       description: Platto aggiornato
119     '404':
120       description: Platto non trovato
121   delete:
122     tags: [Meals]
123     summary: Elimina un piatto
124     parameters:
125       - in: path
126         name: id
127         required: true
128         schema: { type: Integer }
129     responses:
130       '200':
131         description: Platto eliminato
132       '404':
133         description: Platto non trovato
134
135 components:
136   schemas:
137     User:
138       type: object
139       properties:
140         id:
141           type: integer
142           username:
143             type: string
144             email:
145               type: string
146             password:
147               type: string
148             role:
149               type: string
150               enum: [cliente, ristoratore]
151     Meal:
152       type: object
153       properties:
154         idmeal:
155           type: integer
156         restaurantid:
157           type: integer
158         nome:
159           type: string
160           prezzo:
161             type: number
162             format: float
163             tipologia:
164               type: string
135   descriptions: Platto non trovato
136
137 components:
138   schemas:
139     User:
140       type: object
141       properties:
142         id:
143           type: integer
144           username:
145             type: string
146             email:
147             password:
148             type: string
149             role:
150               type: string
151     Meal:
152       type: object
153       properties:
154         idmeal:
155           type: integer
156         restaurantid:
157           type: integer
158         nome:
159           type: string
160           prezzo:
161             type: number
162             format: float
163             tipologia:
164               type: string
165             ingredienti:
166               type: array
167               items:
168                 type: string
169                 imagine:
170                   type: string
171                   format: uri
172
```

Servers
<https://restaurant-management-wzhj.onrender.com>

Users Autenticazione e sessione

POST /login Login con email (o username) e password

POST /register Registrazione nuovo utente (cliente o ristoratore)

POST /users/register Registra un nuovo utente (cliente o ristoratore)

POST /users/login Effettua il login (bcrypt crypt o password in chiaro)

Meals Gestione piatti per ristorante (DB con fallback su file)

GET /meals/common-meals Lista dei piatti comuni (merge DB + file), senza piatti dei ristoranti, con deduplicazione

POST /meals Crea un nuovo piatto (DB, fallback su file)

GET /meals Restituisce i menu per ristorante (DB, fallback su file)

PUT /meals/{id} Aggiorna un piatto per ID (ObjectID o idmeals numerico)

GET /meals/{id} Ottiene un piatto per ID (ObjectID o idmeals numerico)

DELETE /meals/{id} Elimina un piatto (ObjectID o idmeals numerico)

DELETE /meals/{restaurantId}/{id} Elimina un piatto specificando anche il ristorante (compatibilità menu annotato)

database

GET /health Stato server/DB

Orders	
POST	/orders Crea un nuovo ordine
GET	/orders Elenco ordini (filter opzionali)
GET	/orders/{id} Dettaglio ordine (accetta id incrementale o _id Mongo)
PUT	/orders/{id} Aggiorna stato ordine (accetta body con {status} o {state} o {newState})
PUT	/orders/{id}/state Alias per aggiornare lo stato ordine
fetch	
GET	/orders Elenco ordini (filter opzionali)
GET	/restaurants Elenco ristoranti
Restaurants	
GET	/restaurants Elenco ristoranti
POST	/restaurants Crea ristorante
GET	/restaurants/{id} Dettaglio ristorante
PUT	/restaurants/{id} Modifica ristorante per restauranteId
DELETE	/restaurants/{id} Elimina ristorante per restauranteId

Installazione

`npm install --save-dev swagger-autogen`

ulteriori informazioni sono presenti al link sopra riportato

Gestione codici HTTP: codici HTTP sono standard utilizzati per indicare lo stato di una richiesta HTTP effettuata tra un client e un server. Nell'applicazione, vengono ampiamente utilizzati alcuni di questi codici per comunicare lo stato delle richieste e delle risposte:

- Codice 200 (OK): codice di successo. Indica che la richiesta è stata elaborata correttamente e che il server sta restituendo i dati richiesti al client.
- Codice 201(CREATED):la richiesta è andata a buon fine e ha creato una nuova risorsa.
- Codice 204(NO CONTENT):la richiesta è andata a buon fine ma non c'è nessun corpo da restituire.
- Codice 400 (BAD REQUEST): questo codice indica che la richiesta effettuata dal client è stata malformata o non valida. Viene utilizzato quando i dati inviati non corrispondono alle aspettative del server.
- Codice 401 (UNAUTHORIZED): indica che l'accesso a una risorsa richiede l'autenticazione.
- Codice 404 (NOT FOUND): indica che la risorsa richiesta non è stata trovata sul server.
- codice 405(METHOD NOT ALLOWED):il server ha riconosciuto l'URL richiesto, ma non accetta il metodo HTTP usato
- Codice 409(CONFLICT):il server ha ricevuto la richiesta ma non può completarla per un conflitto con lo stato attuale.

- **Codice 500 (INTERNAL SERVER ERROR):** questo codice indica un errore interno del server.
- **Codice 503(SERVICE UNAVAILABLE):** il server non è momentaneamente disponibile

Menù: offerte speciali

Personalized offers only

La pescheria Ristorante

 Cajun spiced fish tacos – (Seafood) € 10.00 Add to cart
 Baked salmon with fennel & tomatoes – (Seafood) € 10.00 Add to cart
 Nasi lemak – (Seafood) € 10.00 Add to cart
 Grilled Portuguese sardines – (Seafood) € 10.00 Add to cart
 Fish fofos – (Seafood) € 10.00 Add to cart
 Escovitch Fish – (Seafood) € 10.00 Add to cart

La crème Ristorante

 Apple Frangipan Tart Dessert Ingredients: digestive biscuits, butter, Bramley apples, Salted Butter, caster sugar, eggs, flour, ground almonds, almond extract, flaked almonds € 10.00 Add to cart	 Apam balik Dessert Ingredients: Milk, Oil, Eggs, Flour, Baking Powder, Salt, Unsalted Butter, Sugar, Peanut Butter € 10.00 Add to cart	 Banana Pancakes Dessert Ingredients: Banana, Eggs, Baking Powder, Vanilla Extract, Oil, Pecan Nuts, Raspberries € 10.00 Add to cart
---	---	--

la pescheria Ristorante

 Cajun spiced fish tacos Seafood Ingredients: cajun, cayenne pepper, white fish, vegetable oil, flour tortilla, avocado, Tomatoes, Salsa, Spring Onions, salsa, sour cream, lemon, garlic € 10.00 Add to cart	 Baked salmon with fennel & tomatoes Seafood Ingredients: Fennel, Parsley, Lemon, Cherry Tomatoes, Olive Oil, Salmon, Black Olives € 10.00 Add to cart	 Nasi lemak Seafood Ingredients: Coconut Milk, Water, Ginger Paste, Ginger, Bay Leaf, Rice, Eggs, Cumin Seeds, Anchovy Fillet, Vegetable Oil, Onion, Garlic, Shallots, Chili Powder, Anchovy Fillet, Sugar, Tamarind Paste € 10.00 Add to cart	 Grilled Portuguese sardines Seafood Ingredients: Sardines, Olive Oil, Garlic, Paprika, Lemon, Rosemary, Red Chili € 10.00 Add to cart	 Fish fofos Seafood Ingredients: Haddock, Potatoes, Green Chili, Coriander, Cumin Seeds, paprika, lime, ginger, flour, eggs, breadcrumbs, vegetable oil € 10.00 Add to cart	 Escovitch Fish Seafood Ingredients: Red Snapper, Vegetable Oil, Garlic, Ginger, Thyme, Bay Leaf, Red Chili, Onion, Carrots, Sugar, Allspice, Worcestershire Sauce, Scotch Bonnet Lime, Malt Vinegar, Pepper € 10.00 Add to cart
---	--	--	--	---	--

Le bistro Ristorante

								
Ayam Percik Chicken	Beef Wellington Beef	Beef Brisket Pot Roast Beef	Beef Sunday Roast Beef	Braised Beef Chilli Beef	Beef stroganoff Beef	Beef Dumpling Stew Beef	Beef and Mustard Pie Beef	Beef Bourguignon Beef
Ingredients: Chicken Thighs, Challots, Garlic, Cane Sugar, Cayenne Pepper, Turmeric, Cumin, Coriander, Fennel, Tamarind Paste, Coconut Milk, Sugar, Water € 10.00	Ingredients: mushrooms, English Mustard, Olive Oil, Beef Fillet, Panna hame, Puff Pastry, Egg, Egg Vocks € 10.00	Ingredients: Beef Brisket, Salt, Onion, Garlic, Thyme, Rosemary, Bay Leaves, Beef stock, Carrots, Mustard, Potatoes € 10.00	Ingredients: Beef, Broccoli, Potatoes, Carrots, plain flour, Eggs, milk, sunflower oil € 10.00	Ingredients: Beef, Onions, Garlic, Olive oil, Chorizo, Cumin, Allspice, Cloves, Cinnamon stick, Bay Leaves, Oregano, Andouille Sausage, Tomato Ketchup, Vinegar, Plum Tomatoes, Tomato Ketchup, Dark Brown Sugar, Bourbon Beans € 10.00	Ingredients: Beef, Onions, Garlic, Plain Flour, Creme Fraiche, Plain Flour, Creme Fraiche, Leek, Swede, Red Wine, Beef Stock, Bay Leaf, Thyme, Parsley, Plain Flour, Baking Powder, Salt, Water € 10.00	Ingredients: Olive Oil, Onions, Garlic, Butter, Mashed Potatoes, Beef Fillet, Plain Flour, Creme Fraiche, English Mustard, Beef Stock, Bay Leaf, Thyme, Parsley, Plain Flour, Baking Powder, Sweet Water € 10.00	Ingredients: Beef, Plain Flour, Rapeseed Oil, Red Wine, Beef Stock, Onion, Carrots, Thyme, Mustard, Egg Yolks, Puff Pastry, Green Beans, Butter, Salt, Pepper € 10.00	Ingredients: Gresse Fat, Beef Shin, Bacon, Challots, Chestnut Mushroom, Garlic Clove, Bouquet Garni, Tomato Puree, Red Wine, Colored Peppers Oil, Thyme, Rosemary, Bay Leaf, Cardamom € 10.00
Add to cart	Add to cart	Add to cart	Add to cart	Add to cart	Add to cart	Add to cart	Add to cart	Add to cart
								
Brown Stew Chicken Chicken	Bitterballen (Dutch meatballs) Beef	BBQ Pork Sloppy Joes Pork						
Ingredients: Chicken, Tomato, Onions, Garlic, Clove, Red Pepper, Carrot, Lime, Honey, Anchovy, Soy Sauce, Cornstarch, Coconut Milk, Vegetable Oil € 10.00	Ingredients: Butter, Flour, Beef Stock, Onion, Parsley, Beef, Salt, Pepper, Nutmeg, Flour, Egg, Breadcrumbs € 10.00	Ingredients: Potatoes, Red Onions, Garlic, Lime, Bread, Pork, Barbecue Sauce, Honey, Tomato Ketchup, Sugar, Vegetable Oil, Salt, Pepper € 10.00						
Add to cart	Add to cart	Add to cart						

Lingua

La scelta di utilizzare la lingua inglese, come standard di programmazione, è ampiamente diffusa nell'industria del software ed è guidata principalmente dal desiderio di aderire allo standard internazionale.

Questo standard è anche noto nella community di programmatore come "English-based programming".

Adottare questa convenzione ha numerosi vantaggi, in quanto rende il codice più leggibile e comprensibile per un pubblico globale di sviluppatori.

La scelta è derivata anche dalla decisione di rendere il progetto, dopo la conclusione, un applicazione open-source.