

# VS COMMERCE

PROGETTO  
TECNOLOGIE INTERNET



# VS COMMERCE

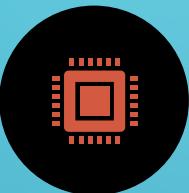


## BREVE PRESENTAZIONE DEL PROGETTO

# TECNOLOGIE UTILIZZATE



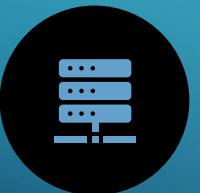
Questo progetto è stato creato utilizzando NodeJS come server, ReactJS come libreria, Express come mini-framework server-side per NodeJS, MySQL per salvare tutti i dati.



Inoltre è stato utilizzato Jsonwebtoken ovvero un Open standard utilizzato per trasmettere informazioni in modo sicuro tra le parti.



Materialize-css come libreria grafica per rendere l'interfaccia utente in stile google.



React-router-dom per permettere di scrivere app single page.



Redux che ci aiuta a rendere lo state prevedibile.



Infine md5 per criptare la password inserite.

# VS COMMERCE

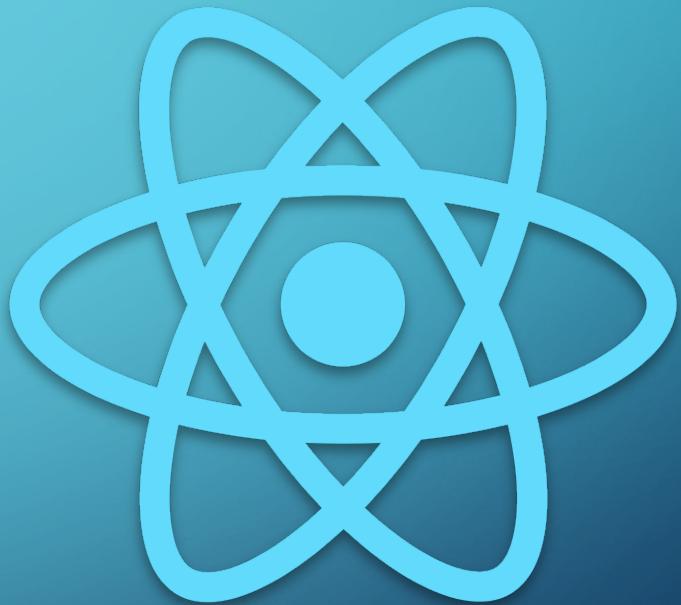


INTRODUZIONE ALLE TECNOLOGIE UTILIZZATE PER  
LA REALIZZAZIONE DEL PROGETTO

# REACT.JS

## Perchè react ?

- React è una LIBRERIA JS e non un framework, creata da Facebook ed utilizzata per creare interfacce utente.
- Abbiamo utilizzato React perchè ci permette di creare la parte front end dell'app, portandoci ad ottenere un interfaccia utente dinamica sia all'interazione dell'utente che al cambiamento dei dati.
- Con React è facile dividere tutte le parti della nostra applicazione in "componenti", quasi atomici e riducendo la complessità a queste singole componenti.
- Utilizzando JS riusciamo a passare informazioni tra i vari componenti e mantenere uno stato.



# NODE.JS



## Perchè NODE ?

- Node come abbiamo studiato, è utilizzato come server per app data-intensive-real-time(DIRT) e quindi è l'ideale per un e-commerce.
- Inoltre dispone di vari moduli/package che possono essere installati per i nostri scopi, come il modulo per la connessione al DB MySQL.
- Node è Event Driven e non-Blocking I/O il che lo rende leggero ed efficiente

# EXPRESS JS

Perche ExpressJS ?

- è un mini-framework server-side per NodeJS, largamente utilizzato che ci permette di creare più velocemente applicazioni in Node, in quanto NodeJS lavora a basso livello.
- Ci permette di creare le nostre REST API in modo semplice ed affidabile.
- Queste API verranno poi chiamate da Client per ottenere i dati che servono all'app.



# MY SQL



## Perchè MySQL ?

- Abbiam deciso di utilizzare MySQL per salvare i nostri dati su Database Relazionale, per avere una sicurezza e affidabilità maggiore rispetto ad altri database.

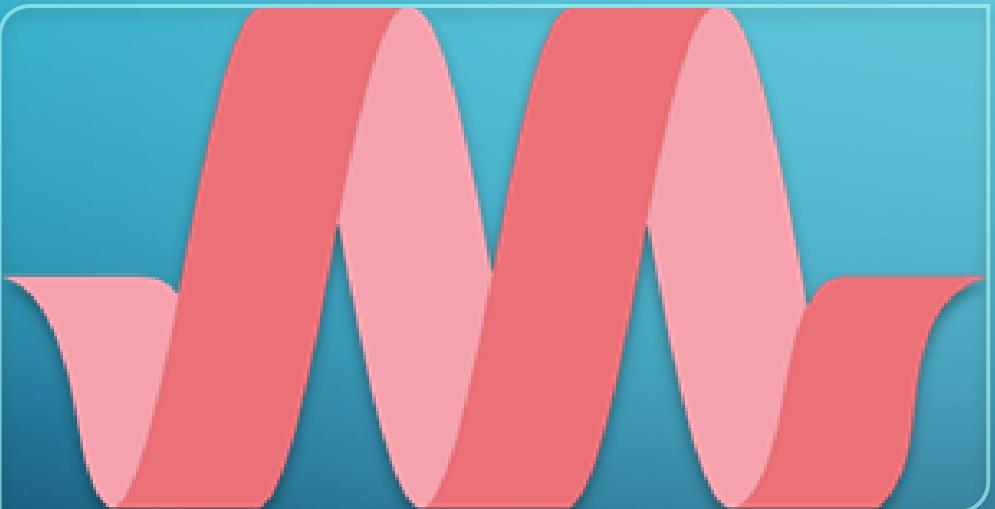
# JSONWEBTOKEN (JWT)

Perchè JWT ?

- Jsonwebtoken è un open standard utilizzato per trasmettere informazioni in modo sicuro tra le parti.
- Lo abbiamo utilizzato per creare un token con una validità di 1 ora, dopo che l'utente si è loggato (in questo modo anche se chiudo il browser e lo riapro l'utente sarà ancora loggato fino ad un ora di tempo, poi automaticamente verrà disconnesso dal suo account).
- Questo token rimane nel Local Storage del Browser e viene verificato dal server quando si va da un pagina all'altra dell'app.
- Il Token è creato e verificato dal server, utilizza un "secret" (segreto) che è presente solo nel server e quindi rende il tutto più sicuro.



# MATERIALIZE-CSS



## Perché Materialize-CSS ?

- Abbiamo scelto di utilizzare la libreria grafica **Materialize-CSS** per rendere l'interfaccia utente molto pulita e semplice, in stile **Google**.

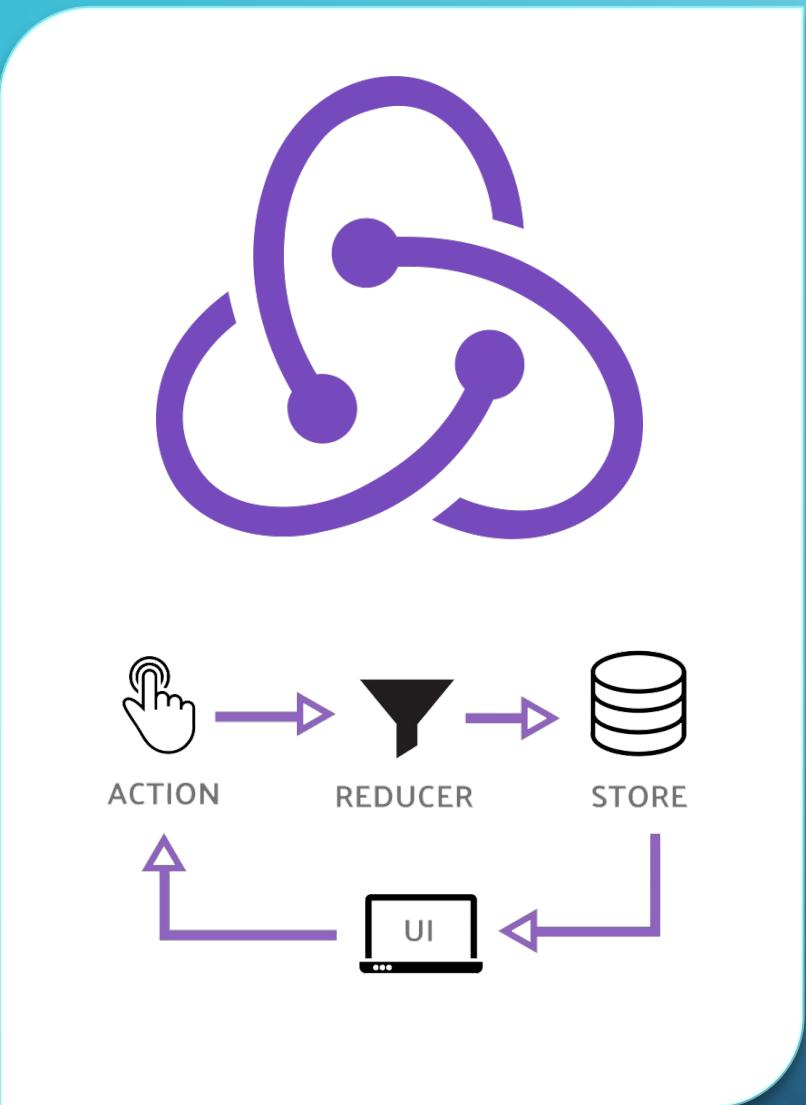
# REACT-ROUTER-DOM

Perché React-Router-Dom ?

- Ci permette di scrivere app single page, invece di ricaricare un intera pagina da server.
- Utilizziamo le Routes e così carichiamo un component diverso in base alla Route.
- Esempio: localhost/admin viene caricato il component admin.



**REACT ROUTER**



# REDUX

Perché Redux ?

- Redux ci aiuta a rendere lo «State» prevedibile.
- Le stesse azione vengono mandate al Reducer e lo stesso risultato ci viene fuori.
- Nel nostro caso avevamo bisogno di avere uno «State» condiviso, in quanto si poteva aggiornare lo stato del carrello da più posizioni (pagina dei prodotti, dettagli del prodotto e carrello stesso).
- Redux ci semplifica questo lavoro, fornendo delle Action che possiamo chiamare per aggiornare il carrello ed avere uno stato condiviso tra vari componenti (che altrimenti viene perso passando da un component ad un altro)

# MD-5

Perché MD-5 ?

- Abbiamo scelto di utilizzare il sistema di cripting MD-5, per criptare le password e avere una sicurezza maggiore per tutti gli utenti del sito



# VSCommerce



ISTRUZIONI DETTAGLIATE PER INSTALLAZIONE E AVVIO DI  
VSCommerce

SCHERMATE E FUNZIONALITÀ DEL SITO

Apache Friends

<https://www.apachefriends.org/it/index.html> ← LINK 1

Scarica Componenti aggiuntivi Hosting Comunità Chi siamo Cerca.. Cerca IT

## XAMPP Apache + MariaDB + PHP + Perl

**2 SCARICARE XAMPP**

**Cos'è XAMPP?**

XAMPP è il più popolare ambiente di sviluppo PHP

XAMPP è una distribuzione di Apache completamente gratuita e semplice da installare, contenente MySQL, PHP e Perl. Il pacchetto open source XAMPP è stato creato per essere estremamente facile da installare e utilizzare.

**Scarica**  
Clicca qui per altre versioni

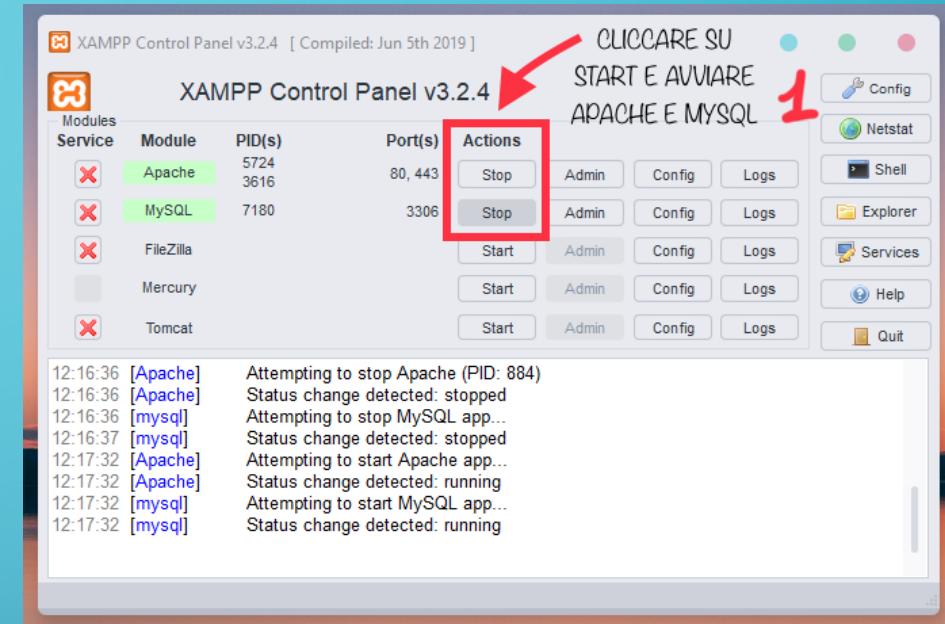
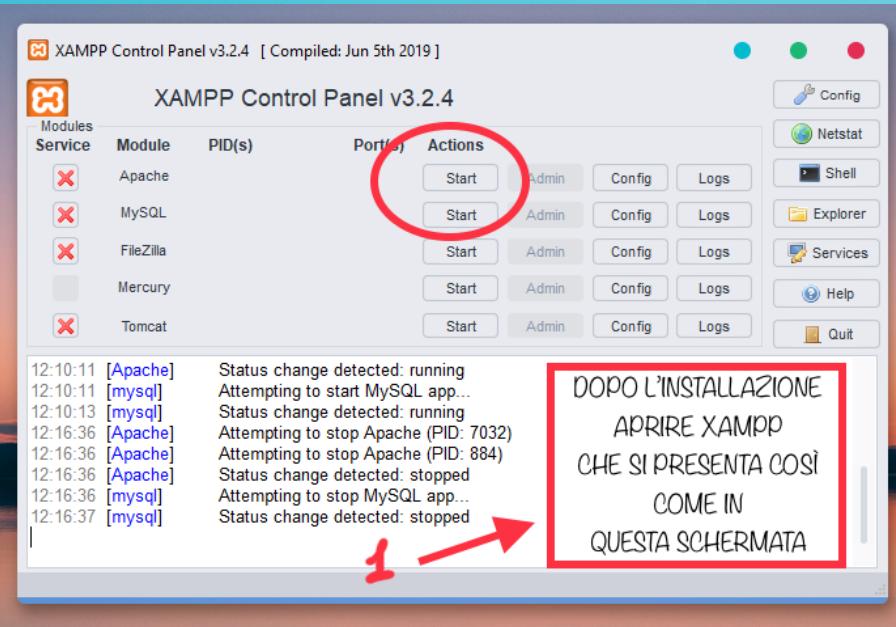
XAMPP per Windows  
7.4.5 (PHP 7.4.5)

XAMPP per Linux  
7.4.5 (PHP 7.4.5)

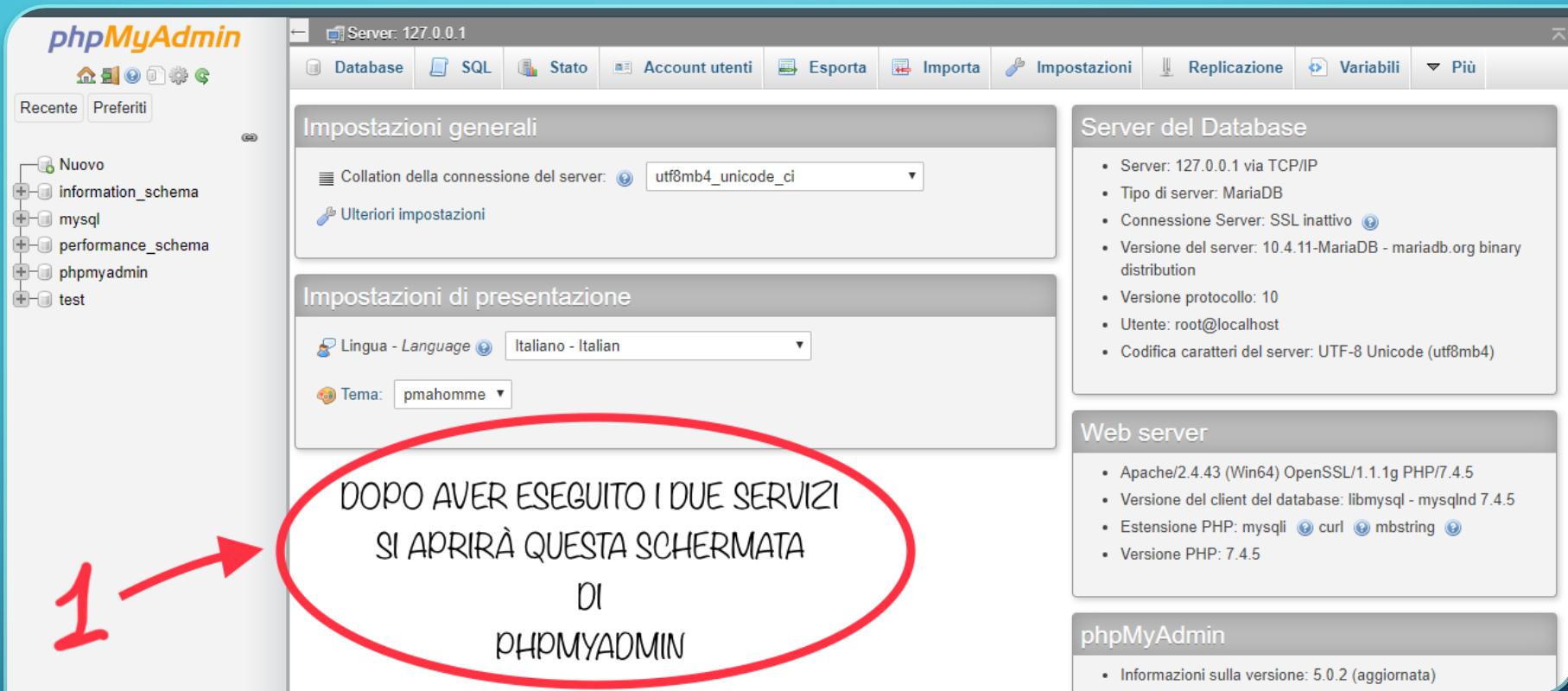
XAMPP per OS X  
7.4.5 (PHP 7.4.5)



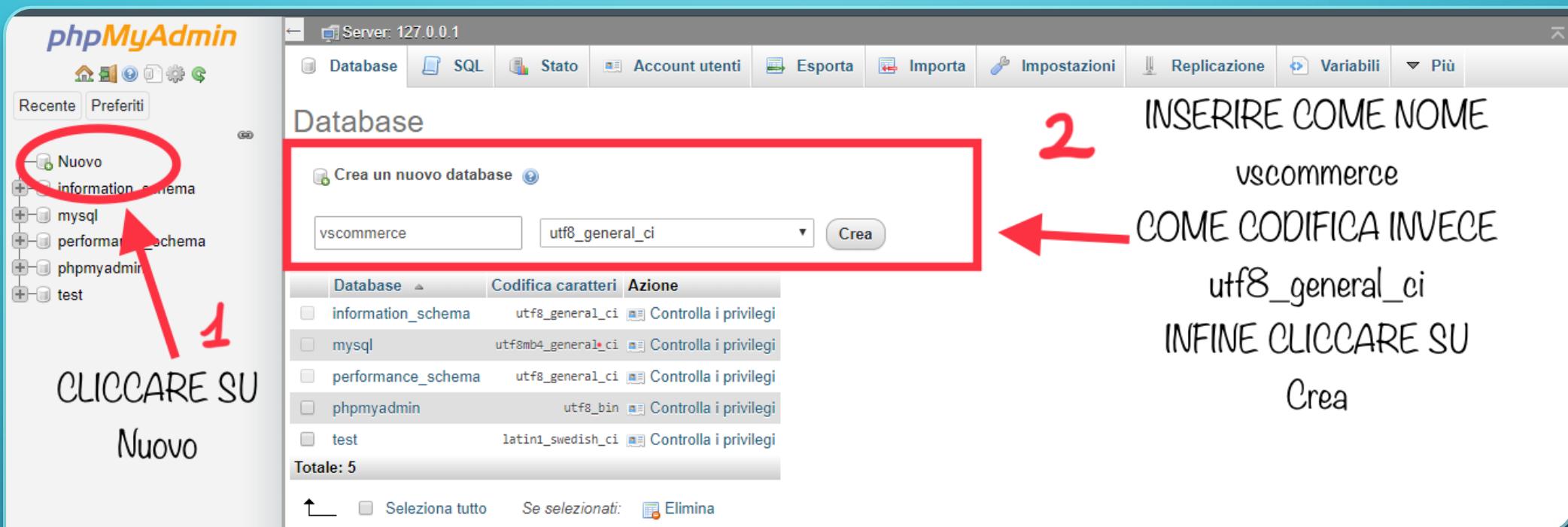
SCARICARE XAMPP  
DOWNLOAD



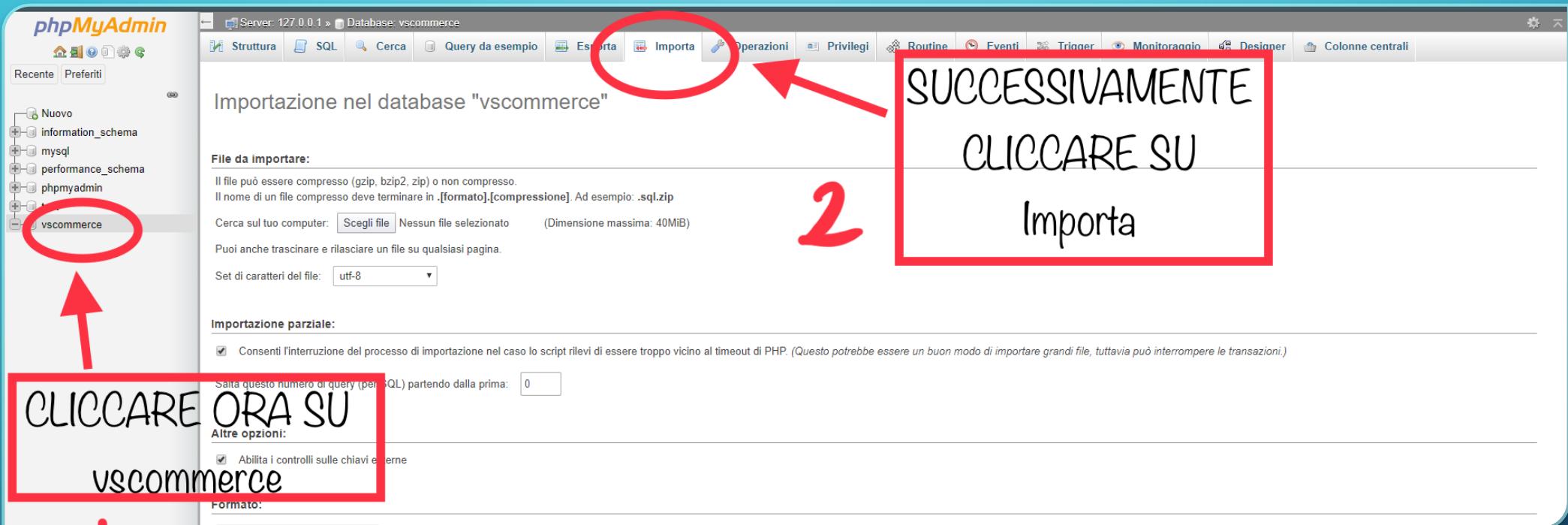
# XAMPP AVVIO



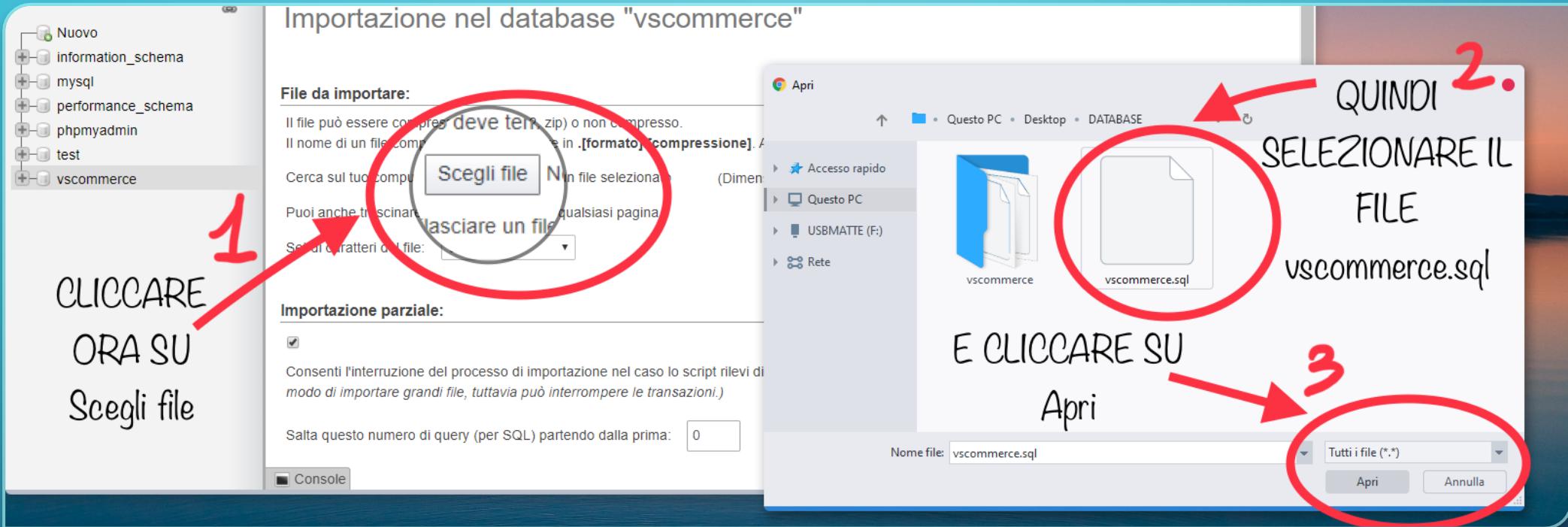
# PHPMYADMIN



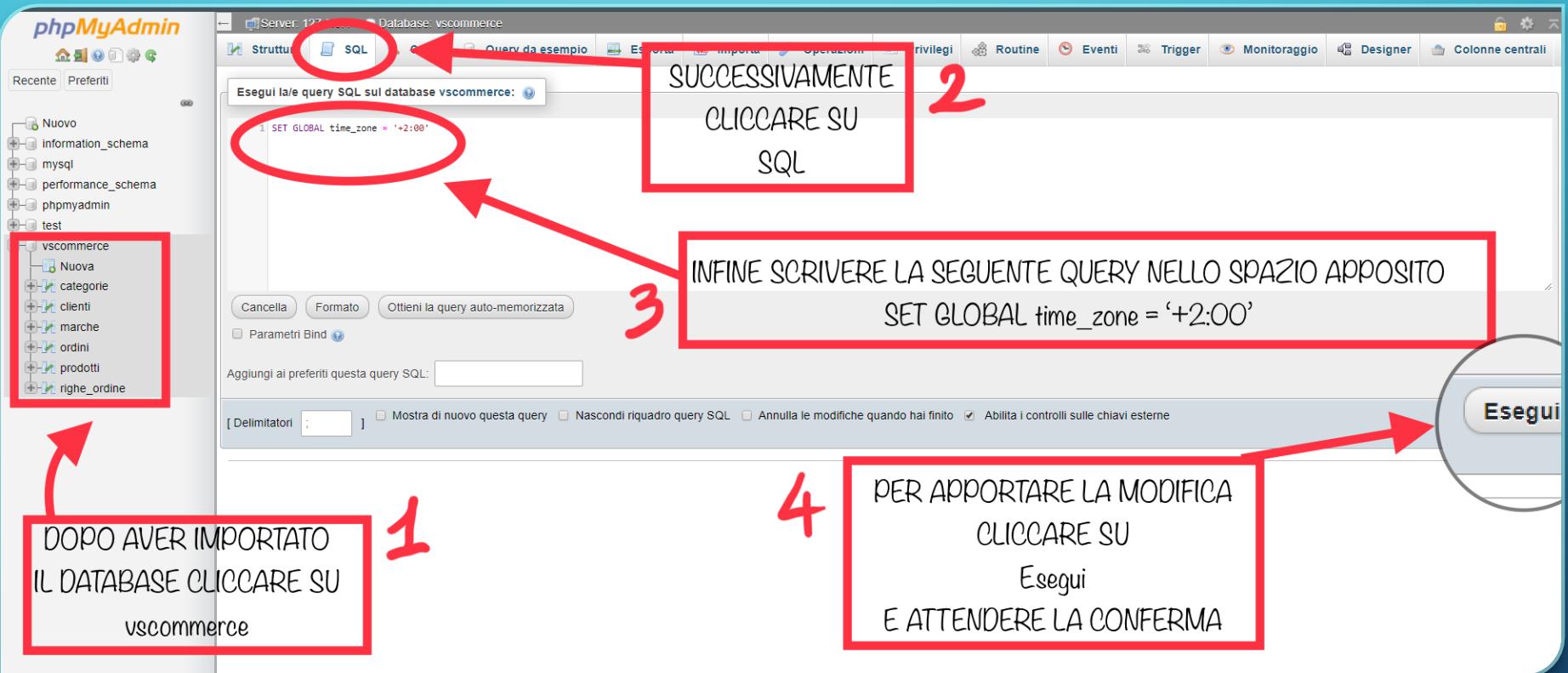
# DATABASE



# IMPORTARE VSCOMMERCE



# VSCOMMERCE



# QUERY SQL

<https://nodejs.org/it/download/>



HOME | INFORMAZIONI | DOWNLOAD | DOCUMENTAZIONE | COME PARTECIPARE | SICUREZZA | BLOG



1

ORA  
INSTALLARE  
Node JS  
RECANDOSI AL  
LINK INDICATO  
QUA E  
SCARICARE LA  
VERSIONE  
NECESSARIA

## Downloads

Latest LTS Version: 12.16.3 (includes npm 6.14.4)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS

Recommended For Most Users

Current

Latest Features



Windows Installer

node-v12.16.3-x64.msi



macOS Installer

node-v12.16.3.pkg



Source Code

node-v12.16.3.tar.gz

Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

Linux Binaries (x64)

Linux Binaries (ARM)

32-bit

64-bit

32-bit

64-bit

64-bit

64-bit

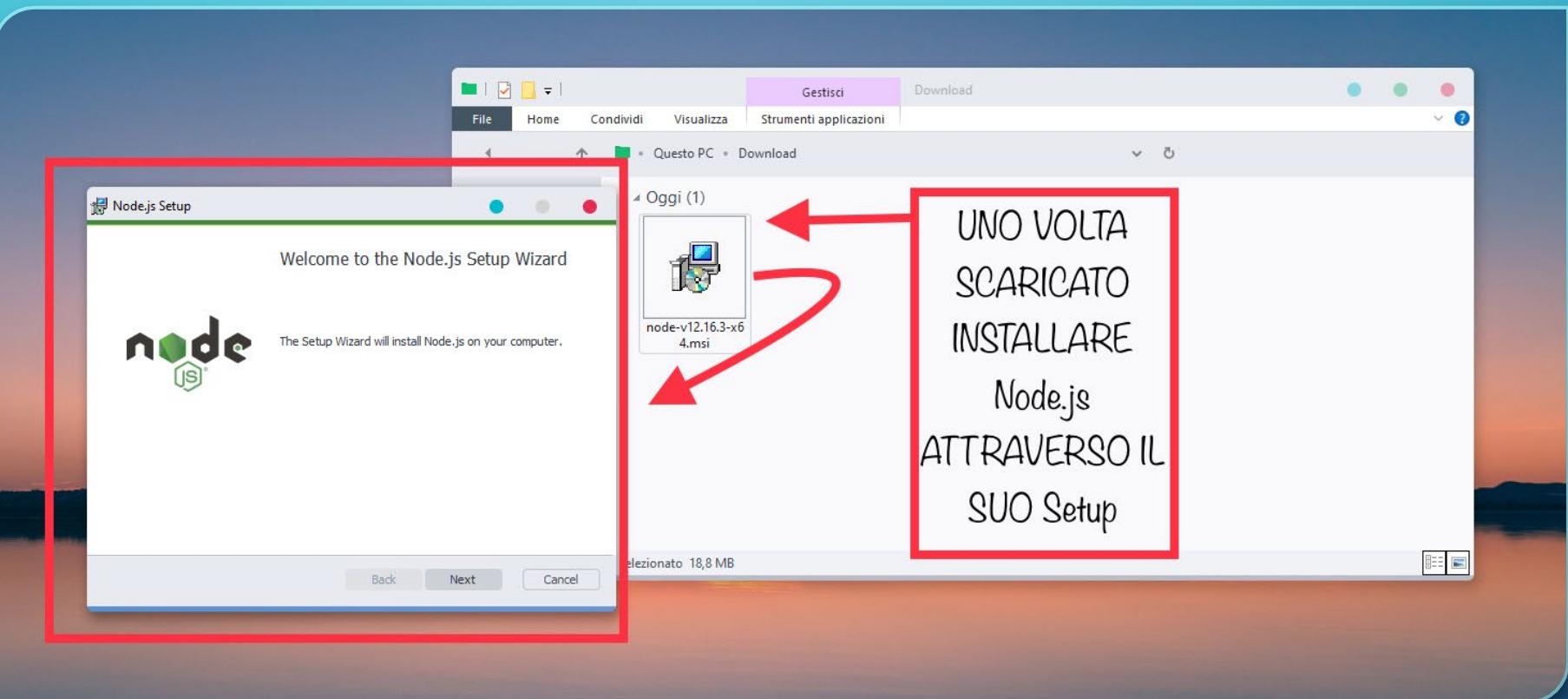
64-bit

ARMv7

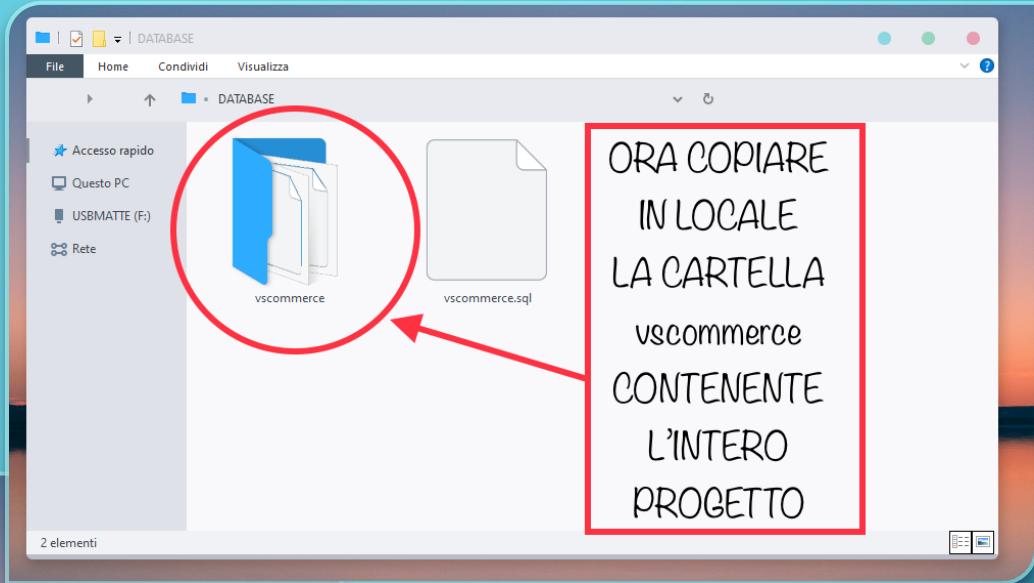
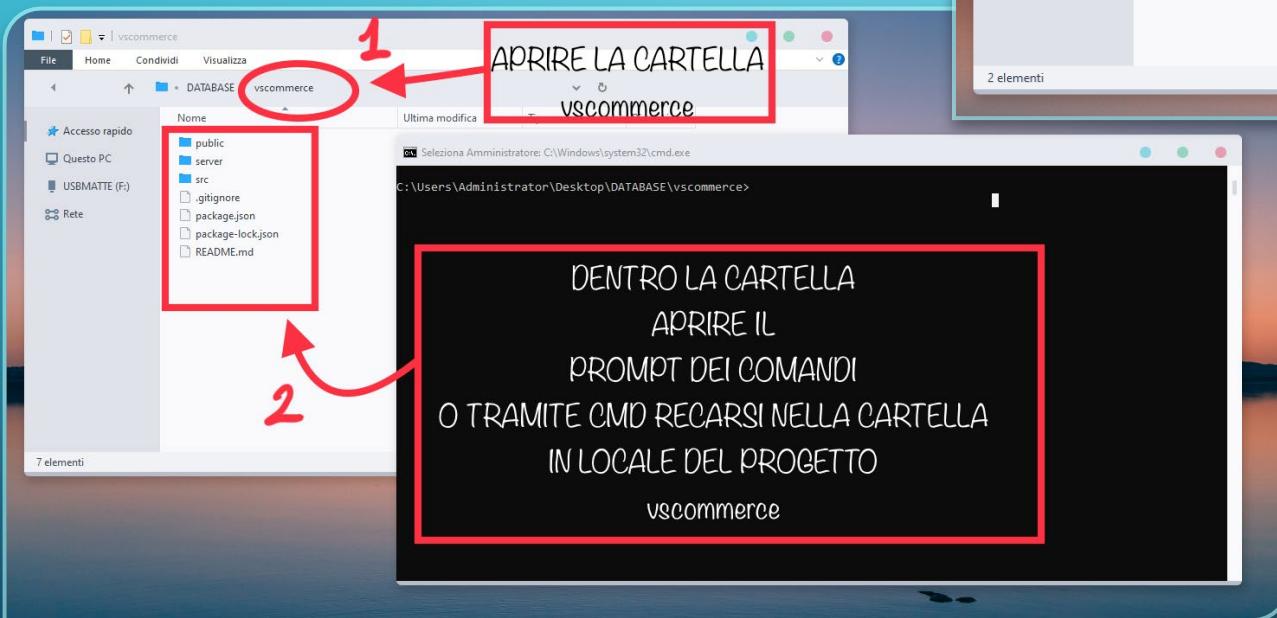
ARMv8

# SCARICARE NODE.JS

## DOWNLOAD



# INSTALLARE NODE.JS



# VSCommerce IN LOCALE

```
C:\ Amministratore: C:\Windows\system32\cmd.exe  
C:\Users\Administrator\Desktop\DATABASE\vscommerce>npm install
```

DAL PROMPT APPENA APERTO  
SCRIVERE ED ESEGUIRE  
IL SEGUENTE COMANDO  
npm install

# PROMPT DEI COMANDI NPM INSTALL

```
> core-js-pure@3.6.5 postinstall C:\Users\Administrator\Desktop\DATABASE\vscommerce\node_modules\core-js-pure
> node -e "try{require('./postinstall')}catch(e){}"

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\webpack-dev-server\node_modules\fsevents):npm WARN notsup SKIPPING OPTIONAL DEPENDENCY:
Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\watchpack\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.12 (node_modules\jest-haste-map\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.12: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

added 1680 packages from 811 contributors and audited 1882 packages in 96.074s

58 packages are looking for funding
  run `npm fund` for details

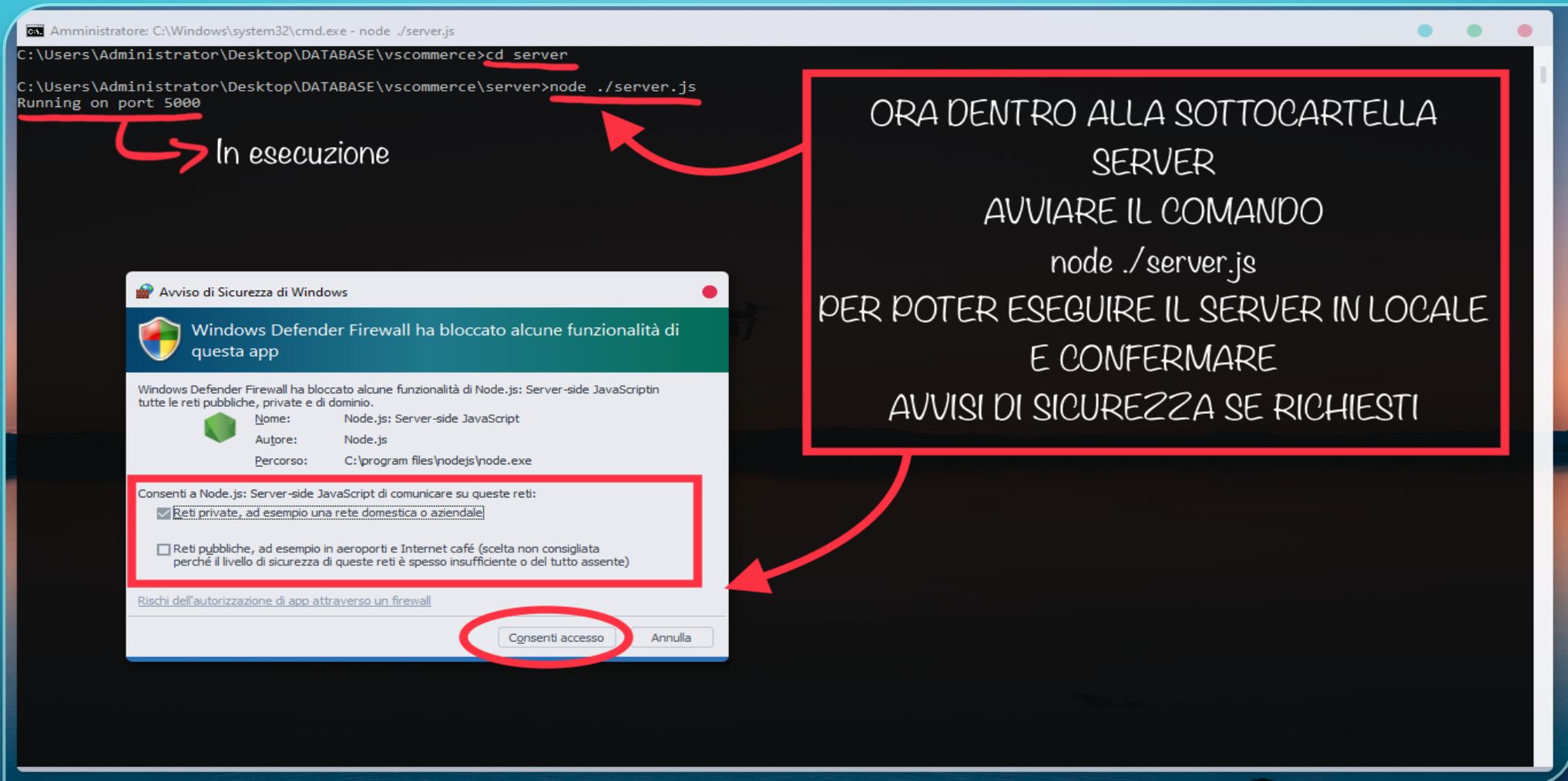
found 3 vulnerabilities (1 low, 2 moderate)
  run `npm audit fix` to fix them, or `npm audit` for details

C:\Users\Administrator\Desktop\DATABASE\vscommerce>cd server
```

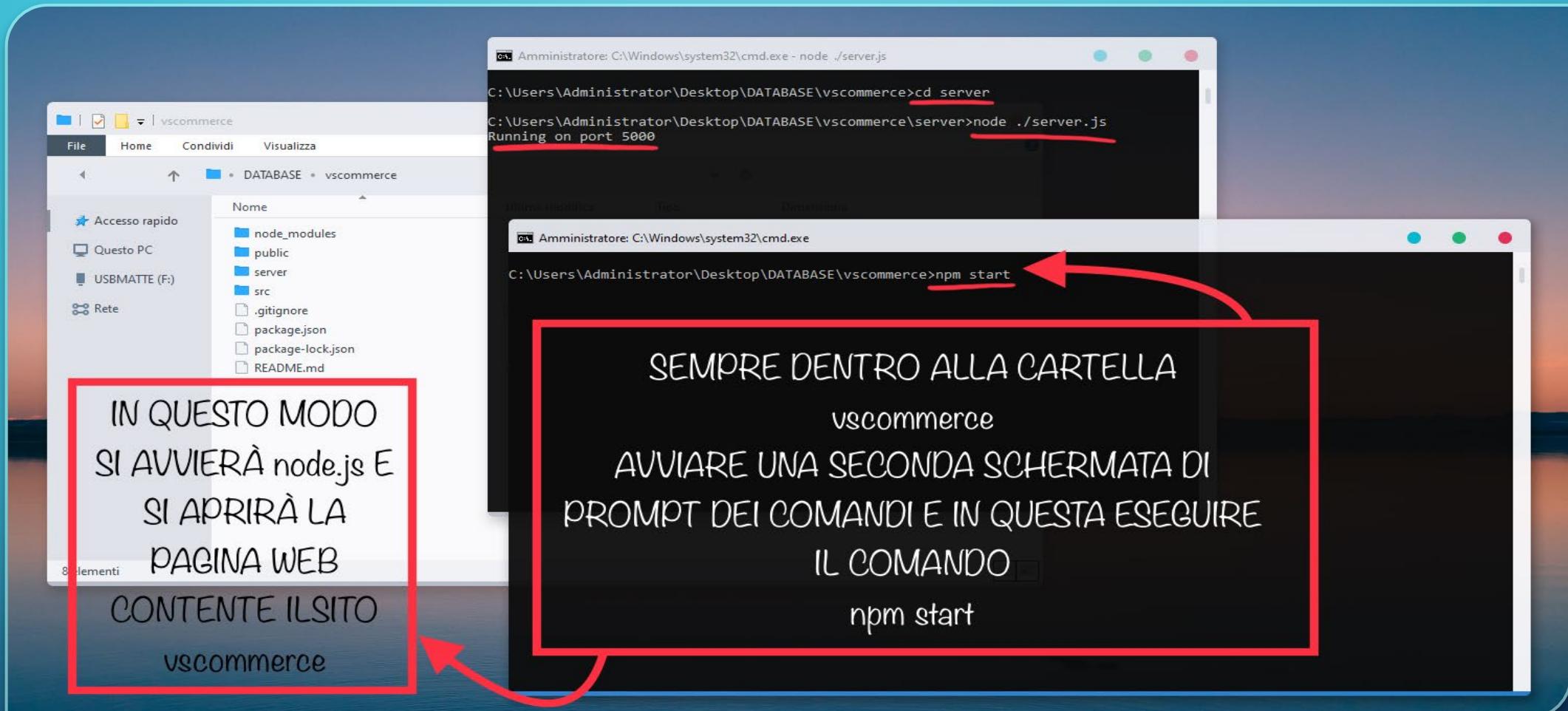
AL TERMINE DELL'INSTALLAZIONE  
SE TUTTO È ANDATO A BUON FINE  
ESEGUIRE IL COMANDO

cd server

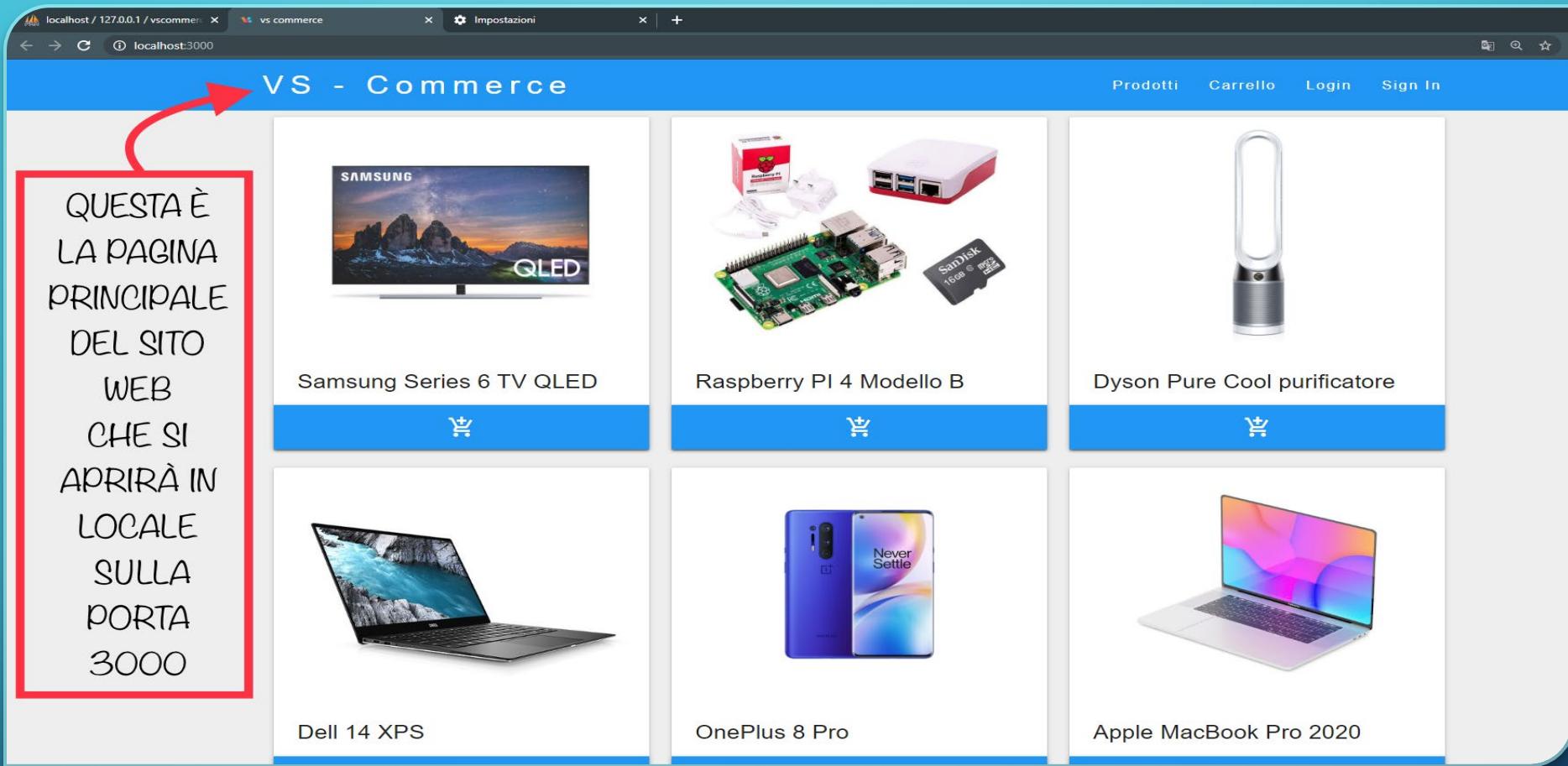
# COMANDO CD SERVER



# COMANDO NODE ./SERVER.JS



# COMANDO NPM START



# AVVIO SITO WEB VS COMMERCE

Dalla pagina di LOGIN sarà possibile accedere al proprio account personale. In base ai privilegi forniti si potrà accedere come UTENTE standard o come AMMINISTRATORE del sito

Email

Password

LOGIN >

Da questa pagina ci si può registrare al sito come UTENTE STANDARD inserendo i dati che verranno salvati nel DATABASE per poter eseguire una LOGIN successivamente

#### Credenziali di accesso

Email

Password

Ripeti password

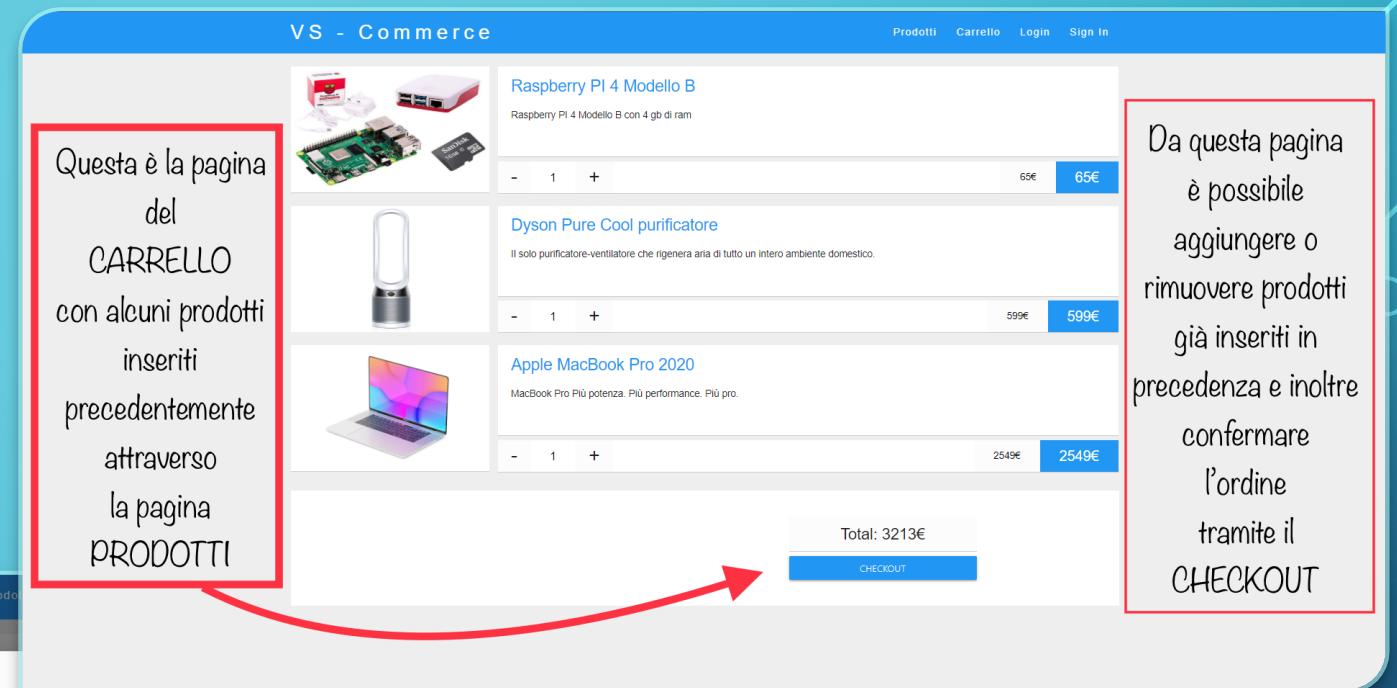
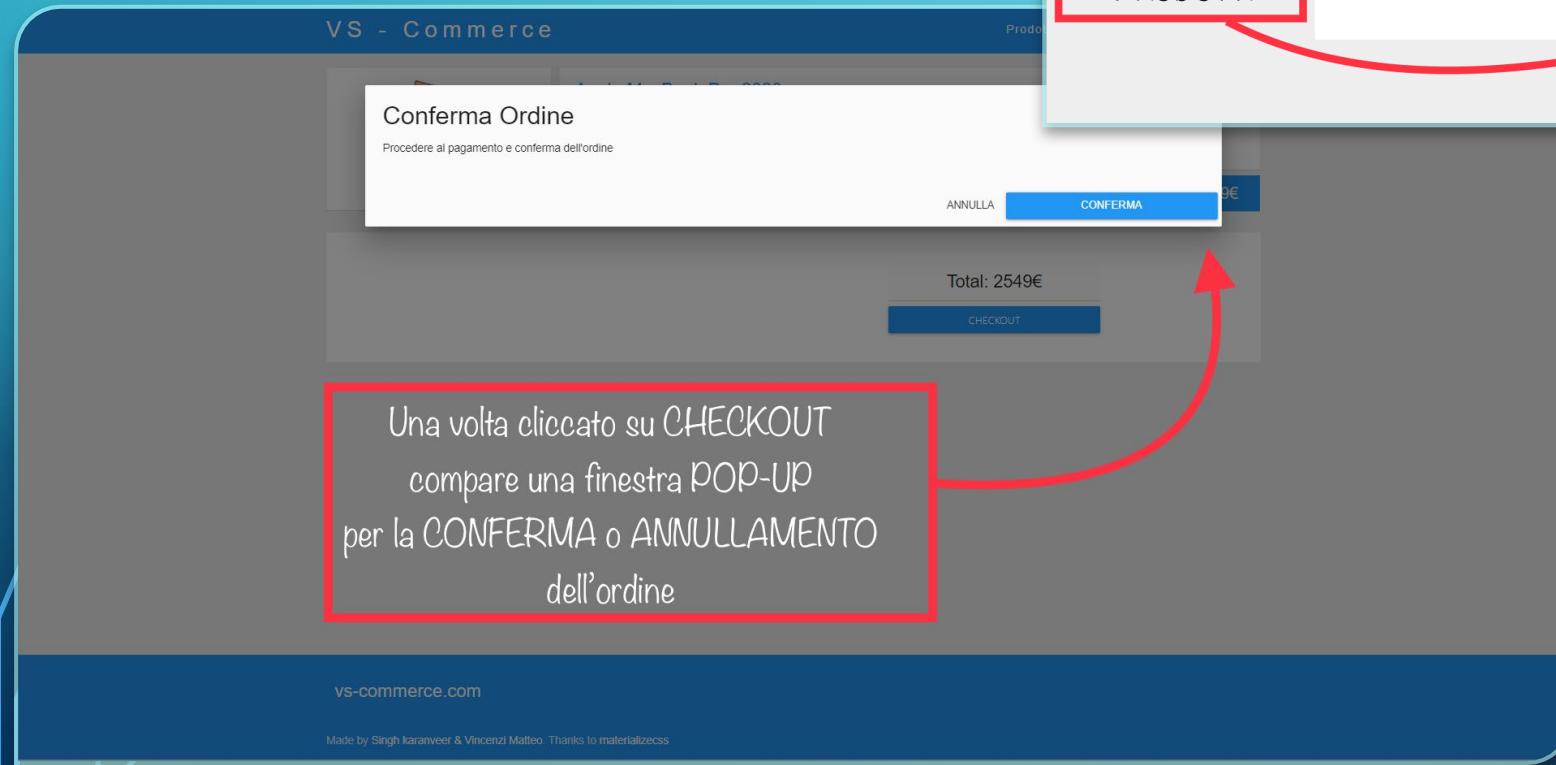
Nome

Cognome

qq/mm/aaaa

Codice fiscale

AVANTI >



Da questa pagina è possibile aggiungere o rimuovere prodotti già inseriti in precedenza e inoltre confermare l'ordine tramite il CHECKOUT

Una volta cliccato su CHECKOUT compare una finestra POP-UP per la CONFERMA o ANNULLAMENTO dell'ordine

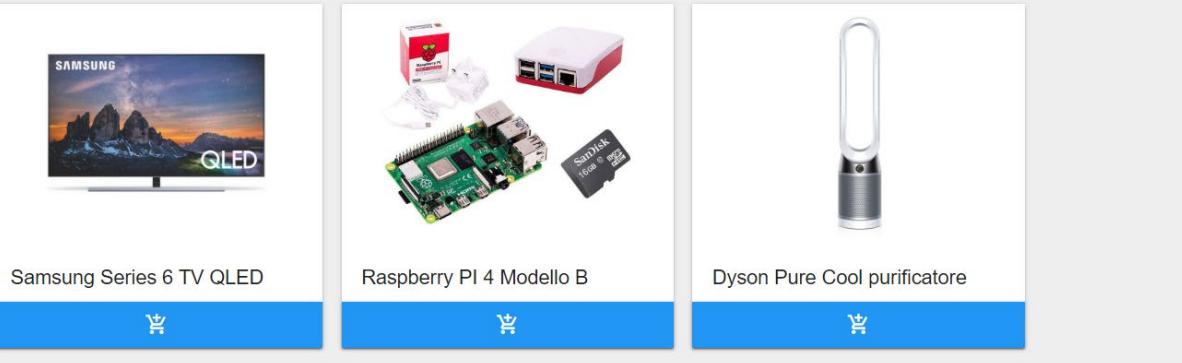
MARCA

prezzo da

fino a

CATEGORIA

AVANTI



La pagina dei PRODOTTI  
nella vista di un utente  
REGISTRATO

Una pagina di  
SIGN IN  
completata con i  
dati  
per la creazione  
di un  
utente standard

## Credenziali di accesso

Email  
user@gmail.com

Password  
\*\*\*\*\*

Ripeti password  
\*\*\*\*\*

Nome  
Pippo

Cognome  
Rossi

01/01/1988

Codice fiscale  
HYHDLM87M11A489E

AVANTI

VS - Commerce

Prodotti Carrello Ordini Account Logout

APPLE  
prezzo da  
500  
fino a  
3000  
CATEGORIA  
AVANTI

Utilizzo del FILTRO per la ricerca specifica dei prodotti

vs-commerce.com

Made by Singh karanveer & Vincenzi Matteo. Thanks to materializecss

Apple MacBook Pro 2020

Apple iPad Pro

iPhone 11 Pro

VS - Commerce

Prodotti Carrello Ordini Account Logout

Ordini

Id Ordine	Data Ordine	Totale
7	May 14 2020 11:38:29	1263 €
6	May 14 2020 11:38:10	2549 €

La pagina degli ORDINI per un utente REGISTRATO

vs-commerce.com

Made by Singh karanveer & Vincenzi Matteo. Thanks to materializecss

Pagina utente  
**AMMINISTRATORE**  
per modifiche alla sezione  
**CATEGORIE**

Nuova categoria

AGGIUNGI >

ABBIGLIAMENTO

ARREDAMENTO

CARTOLERIA

VS - Commerce

Categorie Marche logout

Ordini effettuati: 5

Profitto totale:  
5843.960021972656 €

Prodotto più venduto: QC15  
BOSE[42]

Pagina PRINCIPALE di accesso  
per un utente AMMINISTRAZIONE  
con il riepilogo delle STATISTICHE del sito

vs-commerce.com

Made by Singh karanjeet & Vincenzi Matteo. Thanks to materializecss

Pagina utente  
**AMMINISTRATORE**  
per modifiche alla sezione  
**MARCHE**

Nuova marca

AGGIUNGI >

ANKER

APPLE

BOSE

COLUMBUS

DELL

Pagina utente  
**AMMINISTRATORE**  
per aggiunta di un  
**PRODOTTO NUOVO**

Nome del prodotto

Descrizione

Iva

Prezzo del prodotto

Quantita'

Nome Immagine

Barcode

APPLE

ABBIGLIAMENTO

AGGIUNGI >

Pagina utente  
**AMMINISTRATORE**  
per modifiche alla  
sezione  
**PRODOTTI**



Samsung Series 6 TV QLED  
Euro: 599  
Iva: 22  
Quantita': 85  
Barcode: 5163653



Dell 14 XPS  
Euro: 1099



Raspberry PI 4 Modello B  
Euro: 65  
Iva: 22  
Quantita': 12  
Barcode:



OnePlus 8 Pro  
Euro: 919



Dyson Pure Cool purificatore  
Euro: 599  
Iva: 22  
Quantita': 118  
Barcode:



Apple MacBook Pro 2020  
Euro: 2549



