

## 1. Introduzione

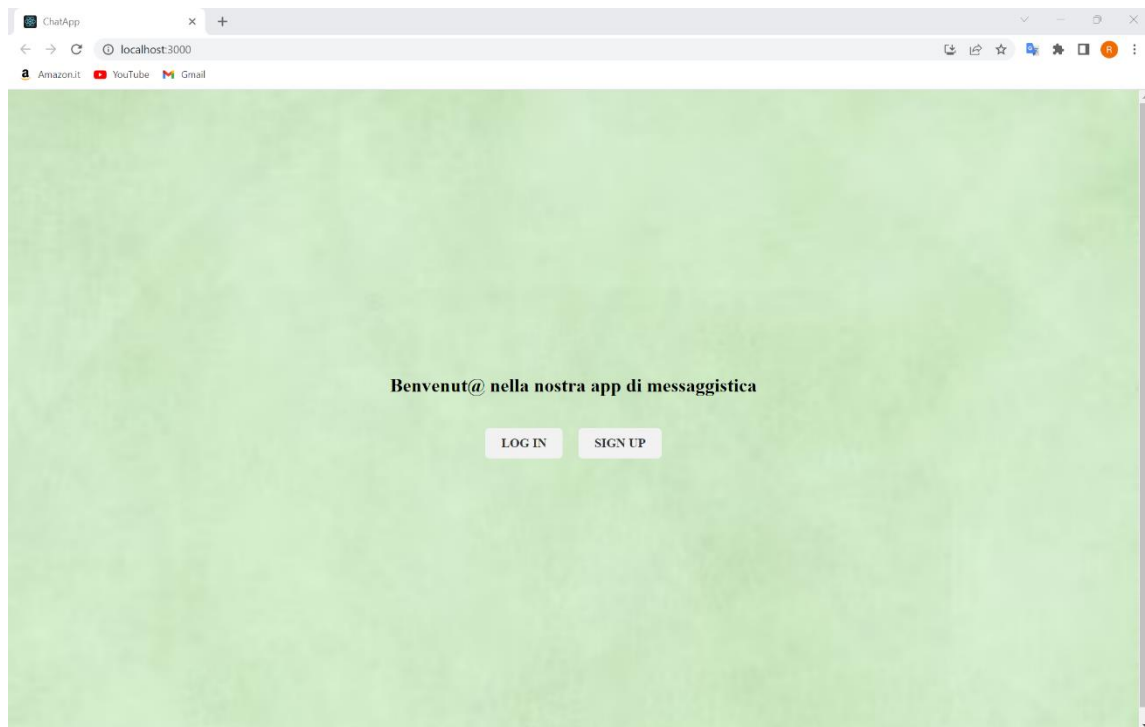
Questo documento ha lo scopo di fornire una guida completa e dettagliata sul funzionamento della nostra applicazione web sviluppata utilizzando lo stack MERN (MongoDB, Express.js, React.js e Node.js) con l'ausilio di Socket.io per la comunicazione real-time. All'interno della nostra applicazione web sono state realizzate le seguenti funzionalità:

- Registrazione sulla piattaforma
- Login
- Visualizzazione messaggi scambiati con un altro utente
- Invio messaggio a un altro utente
- Inserimento di un amico
- Eliminazione di un amico
- Visualizzazione lista amici e accesso alle loro chat
- Interazione real-time della chat (con Socket.IO)
- Uso di fogli di stile CSS e di Material UI

Il codice dell'applicazione è stato arricchito di commenti per favorirne la leggibilità e renderne la comprensione più immediata. Nel documento verranno mostrate le schermate dell'applicazione e spiegato come avviene l'interazione tra app e utente.

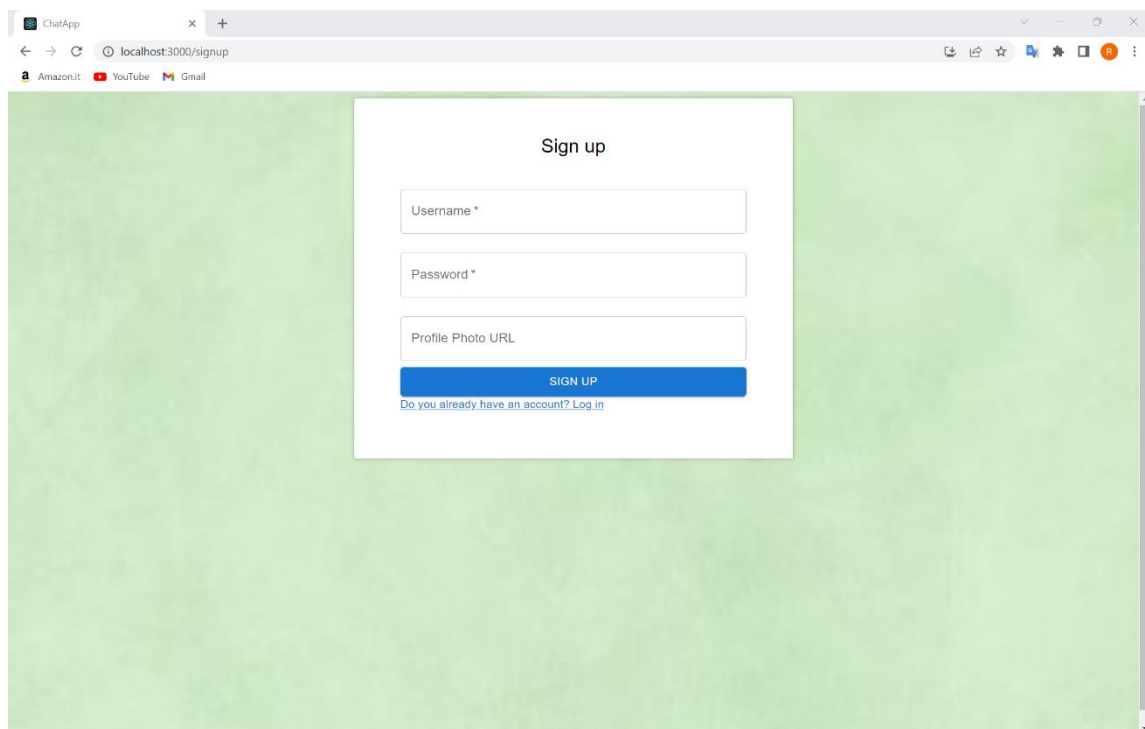
## 2. Funzionamento dell'applicazione

### 2.1 Home



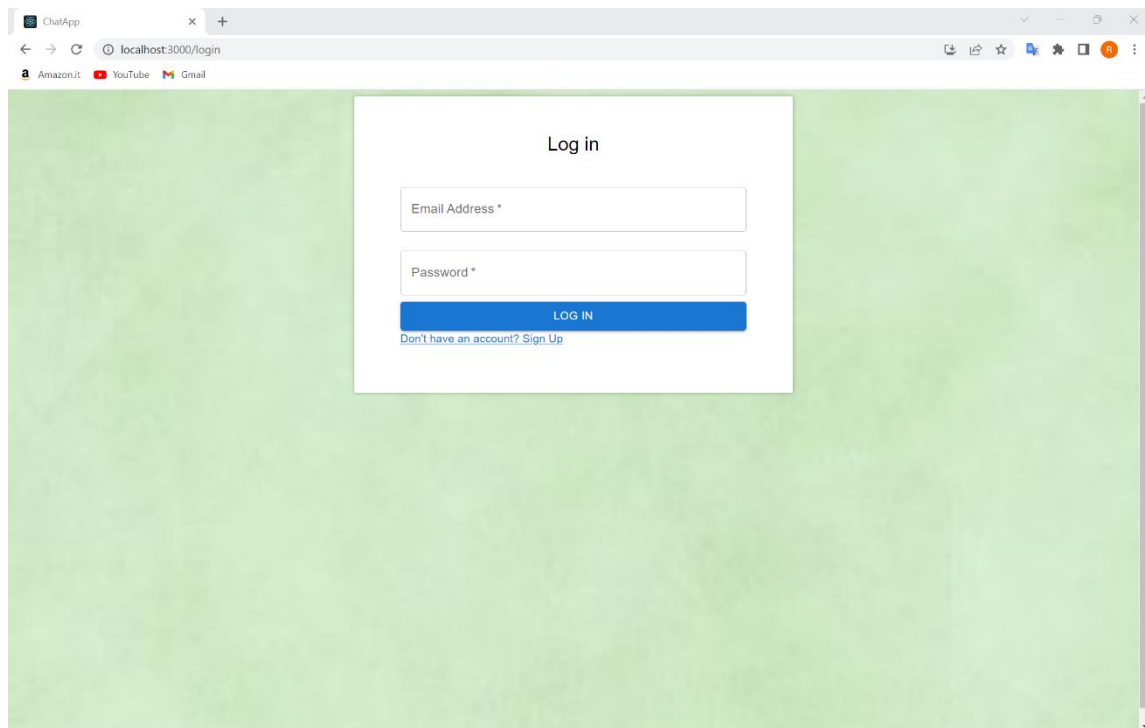
All'avvio dell'applicazione l'utente si troverà a scegliere se effettuare il LOG IN o il SIGN UP

### 2.2 Sign up



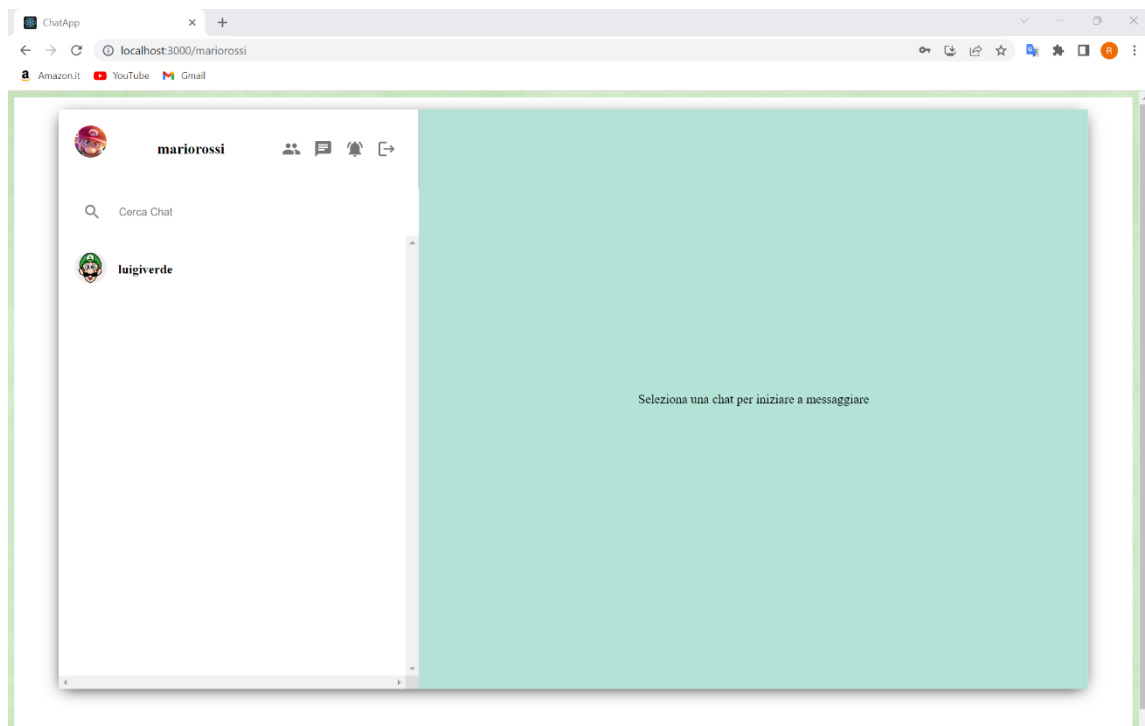
Scegliendo "SIGN UP" l'utente potrà registrarsi inserendo username e password obbligatori, e opzionalmente l'URL di una foto da impostare come immagine di profilo. Nel caso in cui l'username inserito corrisponda a quello di un account già esistente comparirà l'alert di "Utente già esistente".

## 2.3 Log in



Scegliendo “LOG IN” l’utente potrà effettuare l’accesso al suo profilo inserendo username e password. Nel caso in cui l’username inserito non corrisponda a quello di un account già esistente comparirà l’alert di “Utente non trovato”; nel caso in cui l’username inserito corrisponda a quello di un account già esistente ma la password inserita risulta errata comparirà l’alert di “Password errata”.

## 2.4 Homepage dell’utente

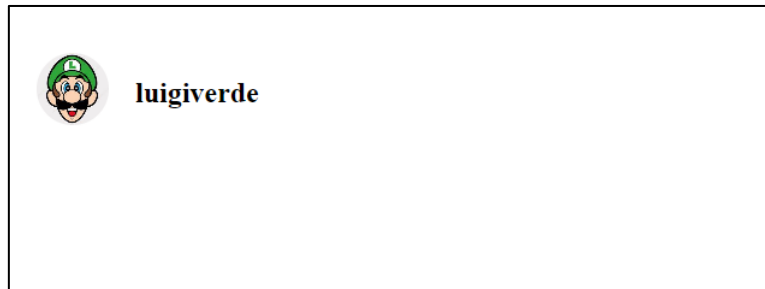


Una volta registrato o aver fatto l’accesso correttamente, l’utente si troverà davanti la seguente schermata. Quest’ultima è composta da una Sidebar alla sinistra e una finestra Chat a destra.



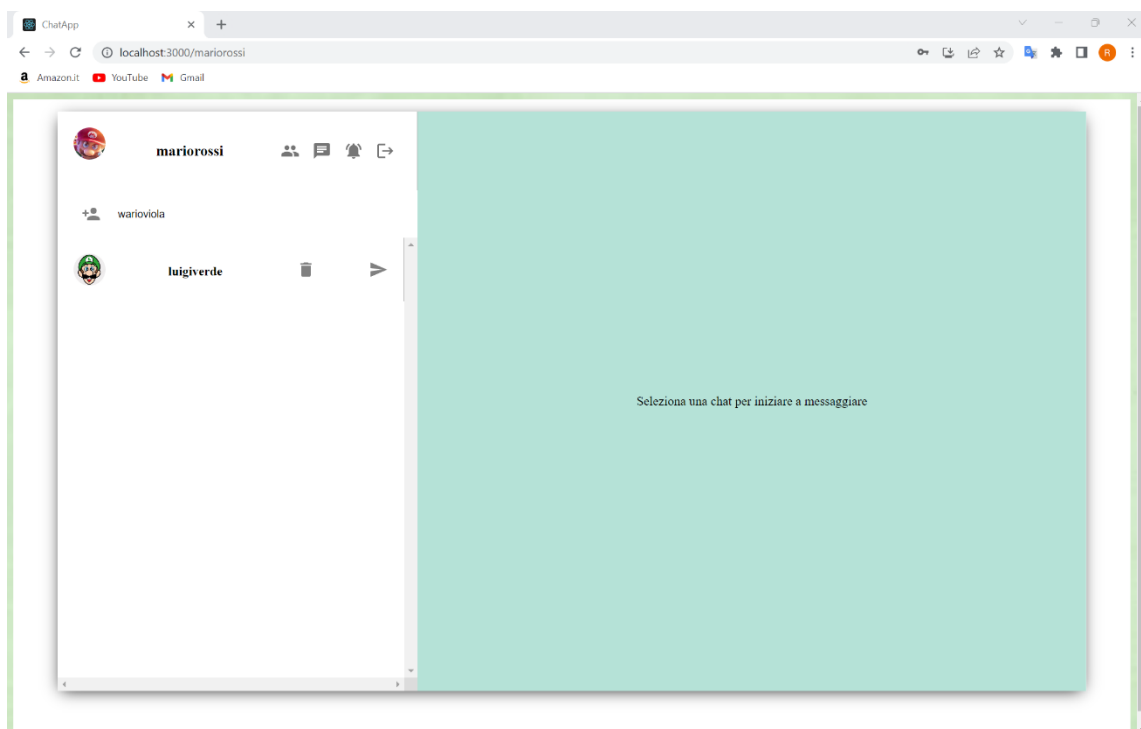
La seguente porzione della Sidebar contiene:

- Immagine di profilo dell'utente loggato
- Username dell'utente loggato
- Icona "PeopleIcon" per visualizzare nella porzione sottostante l'elenco degli amici dell'utente
- Icona "ChatIcon" per visualizzare nella porzione sottostante l'elenco delle chats dell'utente
- Icona "NotificationIcon" per visualizzare nella porzione sottostante l'elenco delle richieste di amicizia dell'utente
- Icona "LogoutIcon" per effettuare il Log out
- La barra di ricerca (elemento inserito per migliorare la grafica ma non gestito)

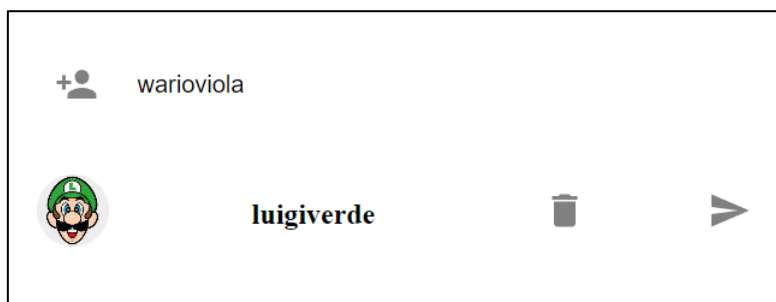


La seguente porzione della Sidebar contiene la lista delle chats dell'utente. Cliccando sulla singola chat verranno visualizzati i messaggi nella finestra Chat, a destra della Sidebar. La lista delle chats viene visualizzata di default nella Sidebar subito dopo il Log in o il Sign up.

## 2.5 Gestione delle amicizie e creazione delle chat



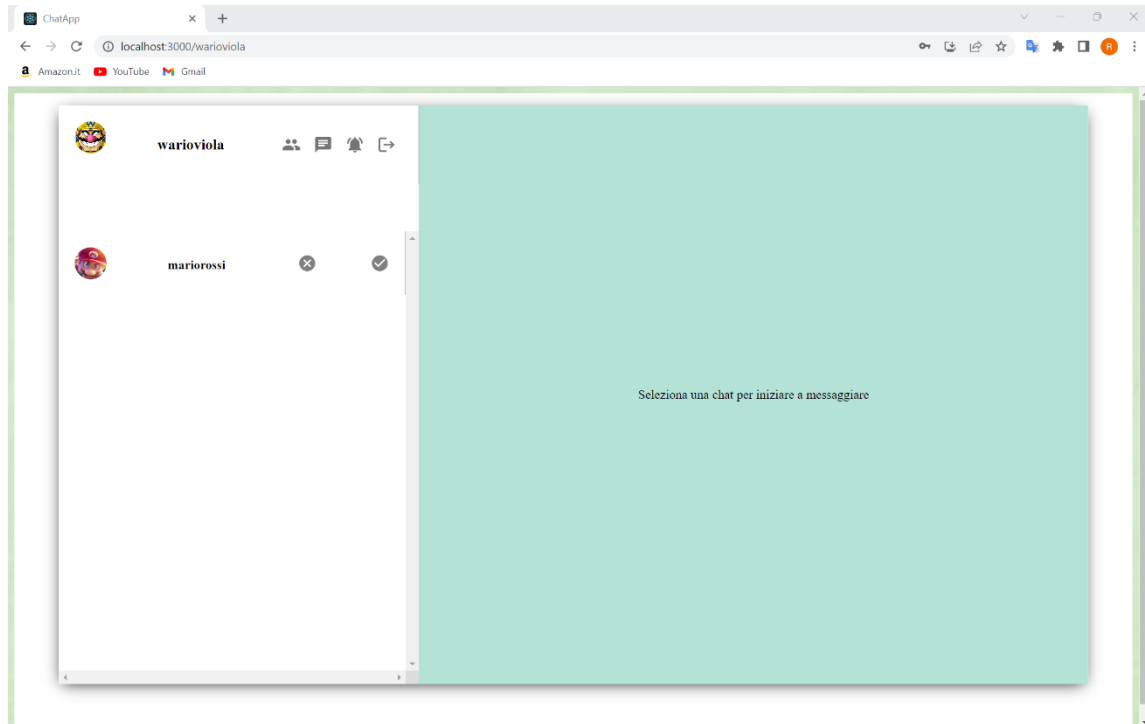
Cliccando sull'icona "PeopleIcon", nella Sidebar verrà visualizzato l'elenco degli amici.



Al posto della barra di ricerca delle chats verrà visualizzata l'icona "PersonAddIcon" e la barra per inviare la richiesta di amicizia ad un utente tramite il suo username (nella foto l'utente sta chiedendo l'amicizia all'utente con username: "warioviola"). L'invio della richiesta di amicizia avviene cliccando sul tasto "Enter" della tastiera. L'invio della richiesta di amicizia non viene consentito se l'utente a cui stiamo inviando la richiesta è già amico dell'utente loggato o se la richiesta è già stata inviata. Nella parte sottostante la barra verranno visualizzati tutti gli amici dell'utente, ciascuno di essi affiancato dalle icone:

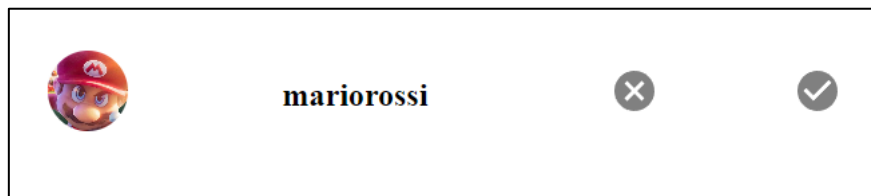
- Icona "DeleteIcon" per l'eliminazione dell'utente dalla lista di amici
- Icona "SendIcon" per la creazione e visualizzazione della chat con l'utente nel caso non esista, e della sola visualizzazione nel caso esista

## 2.6 Gestione delle richieste di amicizia



Siamo nel profilo dell'utente "warioviola" che ha ricevuto la richiesta di amicizia da "mariorossi".

Cliccando sull'icona "NotificationIcon", nella Sidebar verrà visualizzato l'elenco di richieste di amicizia ricevute.

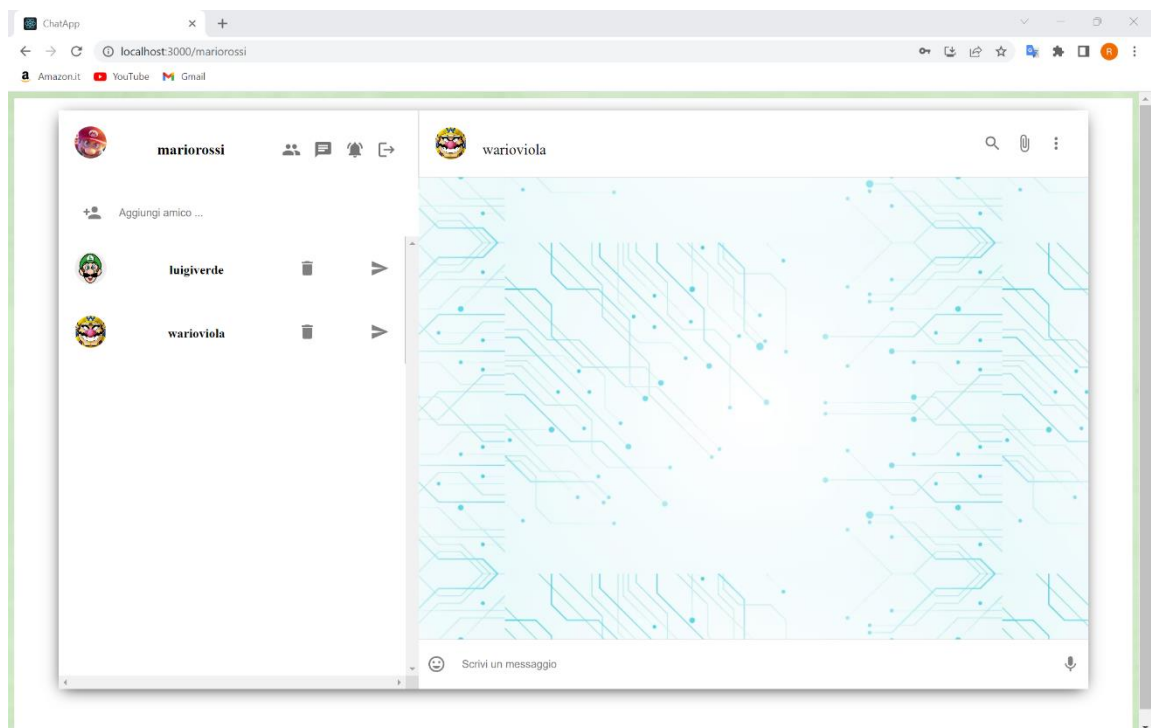
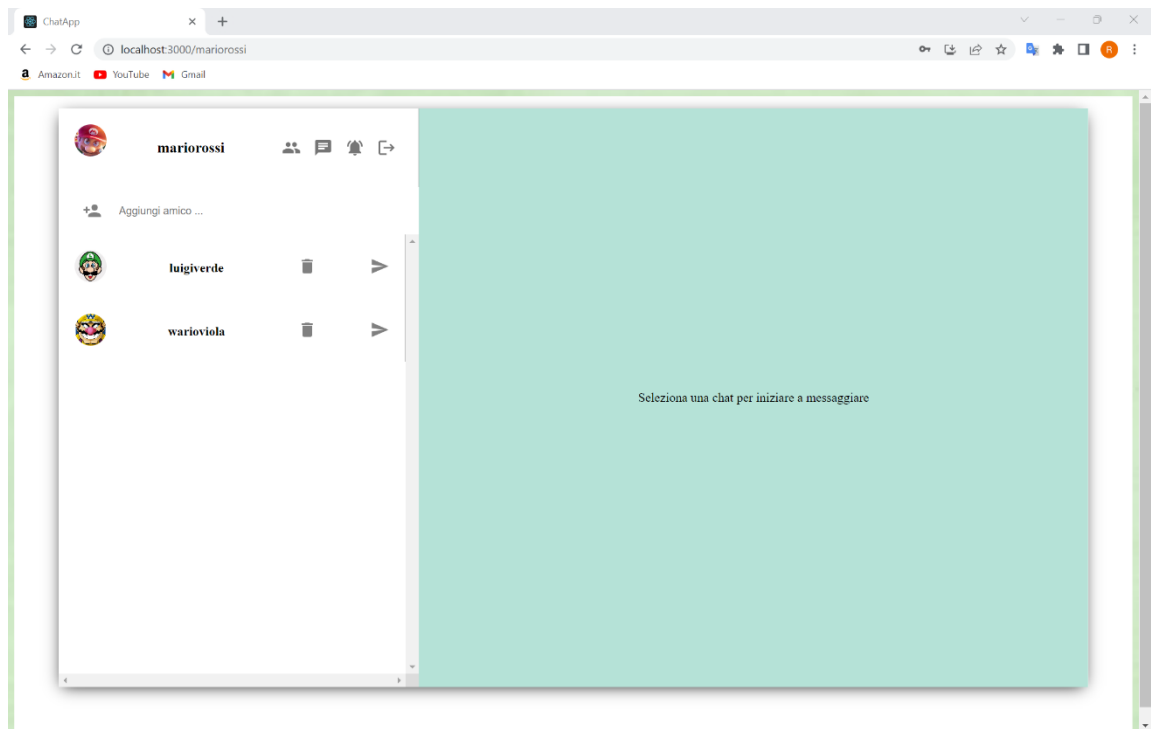


Nella Sidebar, verranno visualizzate tutte le richieste di amicizia ricevute dall'utente loggato, ciascuna di essa sarà affiancata dalle icone:

- Icona "CancelIcon" per rifiutare la richiesta di amicizia
- Icona "CheckCircleIcon" per accettare la richiesta di amicizia.

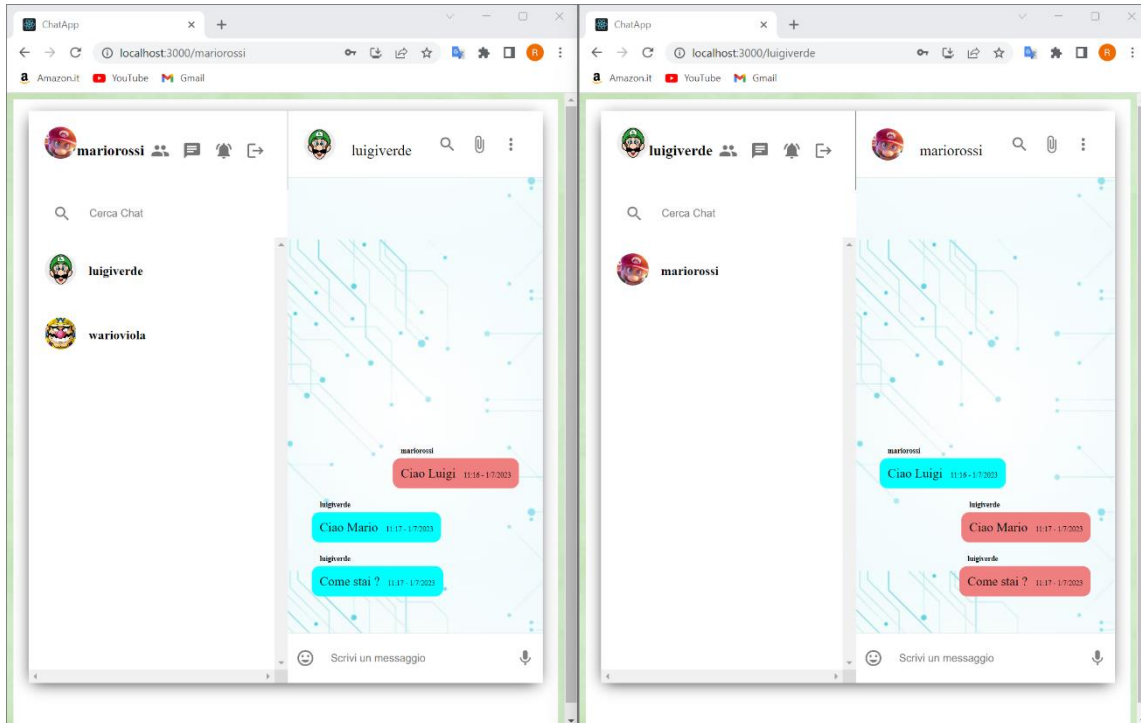
Nel caso in cui la richiesta di amicizia venga accettata, l'utente verrà aggiunto all'interno della lista di amici dell'utente loggato e viceversa.

Ipotizzando che l'utente "warioviola" accetti la richiesta di amicizia di "mariorossi", quest'ultimo visualizzerà l'utente "warioviola" all'interno della sua lista di amici e potrà creare una chat con questo.



## 2.7 Finestra Chat e scambio di messaggi real-time

Ipotizziamo che l'utente loggato "mariorossi" voglia scambiare dei messaggi con l'utente "luigiverde" e che entrambi abbiano aperto la chat. La comunicazione tra i due utenti avverrà in modo sincrono sfruttando l'emissione e intercettazione di eventi all'interno di room in cui questi sono presenti.



All'interno della finestra Chat verranno visualizzati:

- In alto: immagine di profilo e username dell'utente con cui si sta messaggiando e le icone: "SearchIcon", "AttachFileIcon" e "MoreVertIcon" (icone inserite per migliorare la grafica ma non gestite)
- Al centro: elenco dei messaggi scambiati nella chat. In rosso i messaggi mandati dall'utente loggato e in blue i messaggi ricevuti
- In basso: icona "InsertEmoticonIcon", barra di input per l'invio dei messaggi e icona "MicIcon" (icone inserite per migliorare la grafica ma non gestite)

## 2.8 Criticità e punti di miglioramento

La web app riesce a soddisfare tutte le funzionalità descritte nell'introduzione. Durante la fase di testing, abbiamo riscontrato una criticità. Questa criticità consiste nel fatto che qualora un nostro amico decida di rimuoverci dalla sua lista di amici nel momento in cui noi stiamo visualizzando nella finestra Chat i messaggi scambiati con questo, la chat continuerà ad essere visualizzata fino al prossimo render della finestra. Per risolvere questa criticità e migliorare l'interattività dell'applicazione è possibile migliorare o aggiungere alcune funzionalità:

- Estendere l'uso di Socket.IO anche all'invio e ricezione di richieste di amicizia e rimozione di amici
- Prevedere una funzionalità per le icone non gestite
- Utilizzo di cookies per mantenere l'utente loggato in due sessioni successive nel caso in cui non sia stato effettuato il Log out
- Utilizzo di hash key o di token di autenticazione per aumentare la sicurezza