

## RELAZIONE SOCIAL POST-IT

## ITALIANO

I WebSocket possono essere utilizzati nelle API per fornire una comunicazione bidirezionale in tempo reale tra client e server. Ciò significa che, a differenza delle tradizionali API HTTP basate su richieste e risposte, che richiedono che il client faccia una richiesta esplicita al server per ottenere i dati, le API basate su WebSocket consentono ai client di ricevere aggiornamenti dal server in tempo reale, senza dover fare richieste esplicite.

Inoltre, le API WebSocket consentono ai client di inviare dati al server in modo simile, in tempo reale e senza la necessità di fare richieste esplicite.

Le API WebSocket possono essere utilizzate per una vasta gamma di applicazioni, come ad esempio i giochi online, le applicazioni di chat, le applicazioni di collaborazione in tempo reale e le applicazioni di streaming video. Poiché le API WebSocket consentono una comunicazione bidirezionale in tempo reale tra client e server, possono migliorare notevolmente l'esperienza utente e consentire nuove funzionalità che non sarebbero possibili con le API tradizionali basate su HTTP.

Inoltre, molti framework per lo sviluppo di API forniscono supporto per WebSocket, rendendoli facili da integrare nelle applicazioni esistenti.

## INGLESE

WebSockets can be used in APIs to provide real-time two-way communication between client and server. This means that unlike traditional HTTP request-and-response-based APIs, which require the client to make an explicit request to the server to get the data, WebSocket-based APIs allow clients to receive updates from the server in real time, without having to make explicit requests.

Furthermore, the WebSocket APIs allow clients to send data to the server in a similar way, in real time and without the need to make explicit requests.

WebSocket APIs can be used for a variety of applications, such as online gaming, chat applications, real-time collaboration applications, and video streaming applications. Because WebSocket APIs enable real-time two-way communication between client and server, they can greatly enhance the user experience and enable new functionality that wouldn't be possible with traditional HTTP-based APIs.

Additionally, many API development frameworks provide support for WebSockets, making them easy to integrate into existing applications.

# INTRODUZIONE

Il compito assegnato ci chiedeva di creare una pagina web capace di creare dei post-it gestendo la connessione di più client che possono creare e allo stesso momento visualizzare tutti i post-it creati dagli altri client utilizzando i websocket e la libreria Socket.IO.

## CARATTERISTICHE

Il compito è stato impostato creando 4 file per la parte front-end ovvero:

**table.ejs:**

Sarà la pagina dove verrà creata la pagina con all'interno tutti i post-it creati dai client;

**index.ejs:**

Sarà la pagina dove verrà creato il form da compilare per inviare alla pagina table.ejs il nuovo post-it;

**header.ejs:**

E' presente in tutte le pagine e si trova in cima alla pagina e all'interno troviamo il titolo della pagina con un menù a tendina con all'interno i due link per le pagine

/table.ejs e /index.ejs;

**footer.ejs:**

E' presente in tutte le pagine e si trova al fondo della pagina e all'interno troviamo il nome del creatore del progetto;

Mentre per la parte di back-end c'è:

**index.js:**

Si occupa di creare un file JSON se non è già esistente e di richiamare myDepetris.js per leggere dentro il file e poi aggiungere la libreria persona all'interno dei post-it e infine si occupa del reindirizzamento delle pagine;

**myDepetris.js:**

Si occupa di leggere all'interno del file JSON e di scrivere i dati nel file data.JSON.

The image shows a mobile application interface with a dark theme. At the top, there is a black header bar with the word "Home" in white on the left and a white hamburger menu icon on the right. The main content area has a dark blue background with faint, stylized white line art of people's faces. Centered in this area is a white text prompt "Crea il tuo post". Below this prompt are three input fields, each with a light blue icon on the left and a label on the right: the first field has a notepad icon and is labeled "Nickname"; the second field has a person icon and is labeled "Età"; the third field is a larger text area with a speech bubble icon and is labeled "Il tuo post...". At the bottom center of the form is a yellow rectangular button with the word "SUBMIT" in black capital letters.

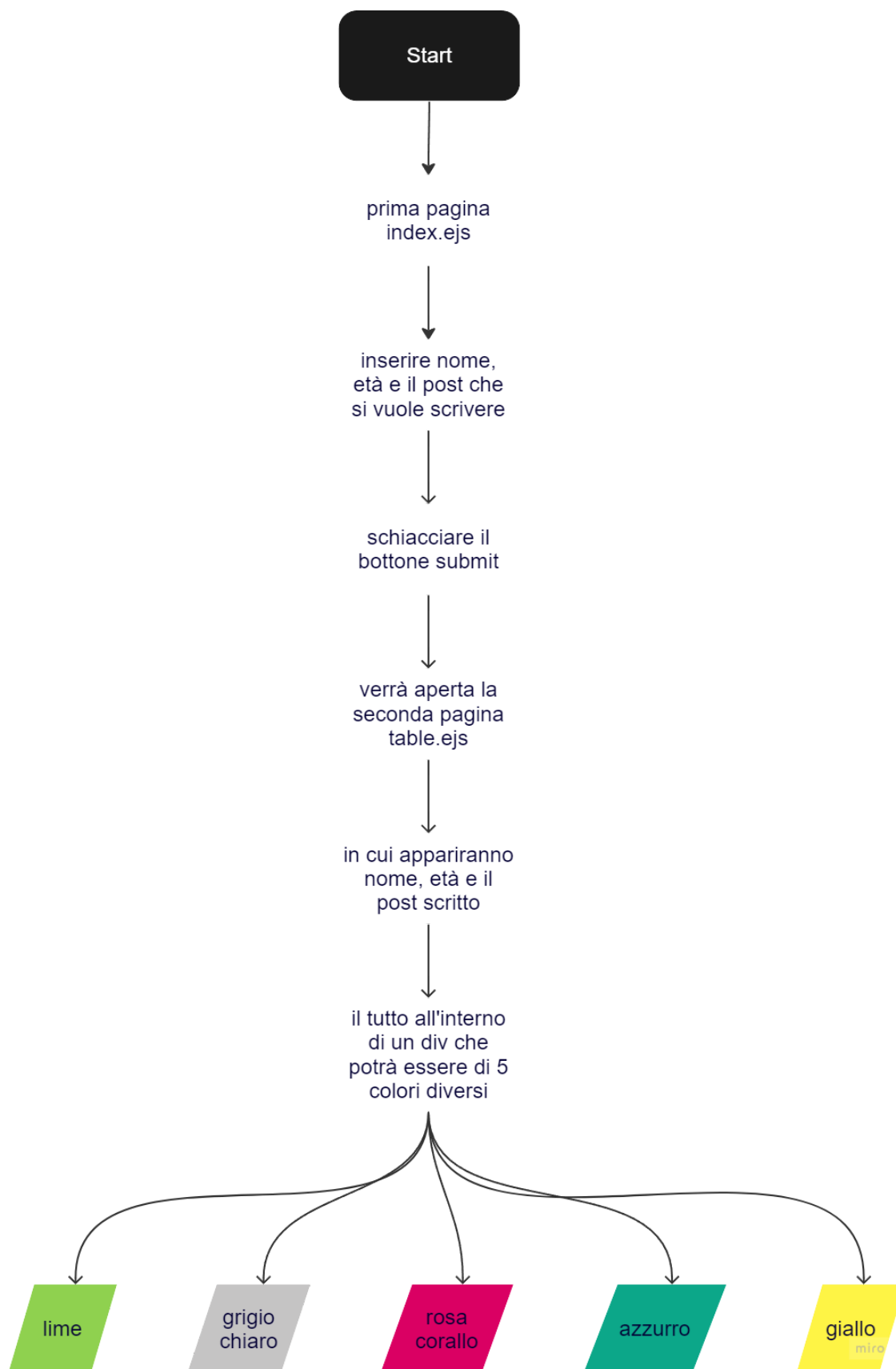
La prima pagina ovvero /index.ejs si presenta con un form centrale con all'interno un titolo "Crea il tuo post" e due text type dove inserire il Nickname e l'età e una Textarea dove poter scrivere quello che di vuole all interno del post-it infine per creare il post-it e visualizzarlo bisogna schiacciare sul pulsante submit.

Table	
<u>Mario Rossi</u> <u>89</u>	Ciao! Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post. Il mio primo post.
<u>Simone</u> <u>19</u>	Ciao iuser si sono proprio io il bimbo della kinder

Nella seconda pagina /table.ejs troviamo dei div che fanno da post-it colorati con all'interno riportati il nome, l'età e un altro div interno bianco con dentro la descrizione; i colori dei post-it vengono dati casualmente ed è possibile che appaia uno di questi 5 colori:

- Verde acqua
  - Lime
  - Giallo
- Grigio chiaro
- Rosa corallo

# DIAGRAMMA DI FLUSSO



## **CRITICITA'**

Il programma potrebbe essere ottimizzato aggiungendo una funzione che permetta di catturare l'ora e la data precisa di quando viene pubblicato un post-it e la locazione geografica di dove lo si pubblica. Inoltre si potrebbe aggiungere un contatore di client connessi nello stesso momento così da sapere quanti utenti sono connessi al sito.

## **CONCLUSIONI**

I WebSocket sono un protocollo di messaggistica che permette una comunicazione asincrona e full-duplex su connessione TCP. I WebSockets non sono connessioni HTTP anche se usano l'HTTP per avviare la connessione. Un sistema full-duplex permette la comunicazione in entrambe le direzioni in maniera contemporanea, tipico esempio di questo tipo di comunicazione è quella telefonica dove gli interlocutori possono parlare ed essere ascoltati allo stesso tempo.

## **Vantaggi nell'uso dei WebSockets**

1. Il WebSocket sono più efficienti e performanti rispetto ad altre soluzioni , come il polling;
2. Richiedono meno banda e riducono la latenza;
3. Semplificano le architetture applicative in real-time;
4. I WebSocket non richiedono intestazione per inviare messaggi riducendo quindi la larghezza di banda richiesta.

[LINK PROGRAMMA](#)