

# “Gioco dei Meme”

VERSIONE FINALE – Le modifiche sono riportate in rosso.

Progettare e implementare un'applicazione web per giocare a una versione single-player del gioco da tavolo “What do you meme?”.

Nel gioco, ogni giocatore riceve un'immagine di un **meme** casuale e sette possibili **didascalie** per quel meme: il giocatore deve indovinare la didascalia “giusta” per il meme.

Il gioco si basa su un archivio di diverse immagini di meme. Ogni immagine di meme è associata ad almeno due didascalie che meglio si adattano all'immagine. È possibile che una didascalia sia associata a più di un meme. Gli studenti sono liberi di utilizzare i meme e le didascalie che preferiscono.

Il gameplay si svolge in più round. Ogni round propone un meme, come segue:

1. Il giocatore riceve un'immagine di un meme casuale e sette possibili didascalie per quel meme in ordine casuale. Sia l'immagine del meme che le didascalie devono essere generate dal server. Tra le sette didascalie, due di esse devono essere tra quelle che meglio si adattano a quel meme.
2. Il giocatore deve selezionare una didascalia che meglio si adatta al meme entro 30 secondi.
3. Se il giocatore seleziona una delle due didascalie più appropriate per il meme entro 30 secondi, ottiene 5 punti e l'applicazione mostra un messaggio che segnala la fine del round.
4. Se il giocatore seleziona una delle altre didascalie o il tempo scade, ottiene 0 punti. In questo caso, l'applicazione mostra un messaggio adeguato insieme alle due didascalie che sono le migliori per il meme dato.

Gli **utenti registrati** (cioè, dopo aver fatto login) giocheranno una partita composta da 3 round. Inoltre, tutte **le loro partite e i loro round saranno registrati al termine di ogni partita**, la cronologia dei loro punteggi sarà visibile in una pagina di profilo utente. **La cronologia deve mostrare tutte le immagini dei meme con il punteggio ottenuto (0 o 5) per round e il punteggio totale della partita.** I meme già mostrati non dovranno essere ripetuti nei **round successivi della stessa partita**.

Dopo 3 round, il gioco si ferma e viene visualizzato il punteggio totale **della partita stessa**, insieme a un riepilogo **che include solo i meme abbinati correttamente e la loro didascalia selezionata**.

Gli **utenti anonimi** (cioè, i visitatori del sito), invece, possono giocare solo partite di un round. Non avranno accesso a nessuna funzionalità degli utenti registrati.

Dopo la fine del gioco, il giocatore può ricominciare una nuova partita.

L'organizzazione di queste specifiche in diverse schermate (e possibilmente su diverse route) è lasciata allo studente.

## Requisiti del progetto

- L'architettura dell'applicazione e il codice sorgente devono essere sviluppati adottando le migliori pratiche (best practice) di sviluppo del software, in particolare per le single-page application (SPA) che usano React e HTTP API.

- L'applicazione deve essere pensata per un browser desktop. La responsività per dispositivi mobile non è richiesta né valutata.
- Il progetto deve essere realizzato come applicazione React, che interagisce con API HTTP implementate in Node.js+Express. La versione di Node.js deve essere quella usata durante il corso (20.x, LTS). Il database deve essere memorizzato in un file SQLite. Il linguaggio di programmazione deve essere JavaScript.
- La comunicazione tra il client ed il server deve seguire il pattern dei "due server", configurando correttamente CORS e con React in modalità "development" con lo Strict Mode attivato.
- La valutazione del progetto sarà effettuata navigando all'interno dell'applicazione. Non saranno testati né usati il bottone di "refresh" né l'impostazione manuale di un URL (tranne /), e il loro comportamento non è specificato. Inoltre, l'applicazione non dovrà mai "autoricarsi" come conseguenza dell'uso normale dell'applicazione stessa.
- La directory radice del progetto deve contenere un file README.md e contenere due subdirectories (client e server). Il progetto deve poter essere lanciato con i comandi: "cd server; nodemon index.mjs" e "cd client; npm run dev". Un template con lo scheletro delle directory del progetto è disponibile nel repository dell'esame. Si può assumere che nodemon sia già installato a livello di sistema. Nessun altro modulo sarà disponibile globalmente.
- L'intero progetto deve essere consegnato tramite GitHub, nel repository creato da GitHub Classroom.
- Il progetto **non deve includere** le directory node\_modules. Esse devono essere ricreabili tramite il comando "npm install" subito dopo "git clone".
- Il progetto può usare librerie popolari e comunemente adottate (per esempio, day.js, react-bootstrap, ecc.), se applicabili e utili. Tali librerie devono essere correttamente dichiarate nei file package.json cosicché il comando npm install le possa scaricare ed installare.
- L'autenticazione dell'utente (login e logout) e l'accesso alle API devono essere realizzati tramite Passport.js e cookie di sessione. Le credenziali devono essere memorizzate in formato hashed e con sale. La registrazione di un nuovo utente non è richiesta né valutata.