A MERN APP

tackchat

Learn More ⟶

tackchat

OUR PROFESSIONAL TEAM

NICOLÒ CLEMENTE

tackchat

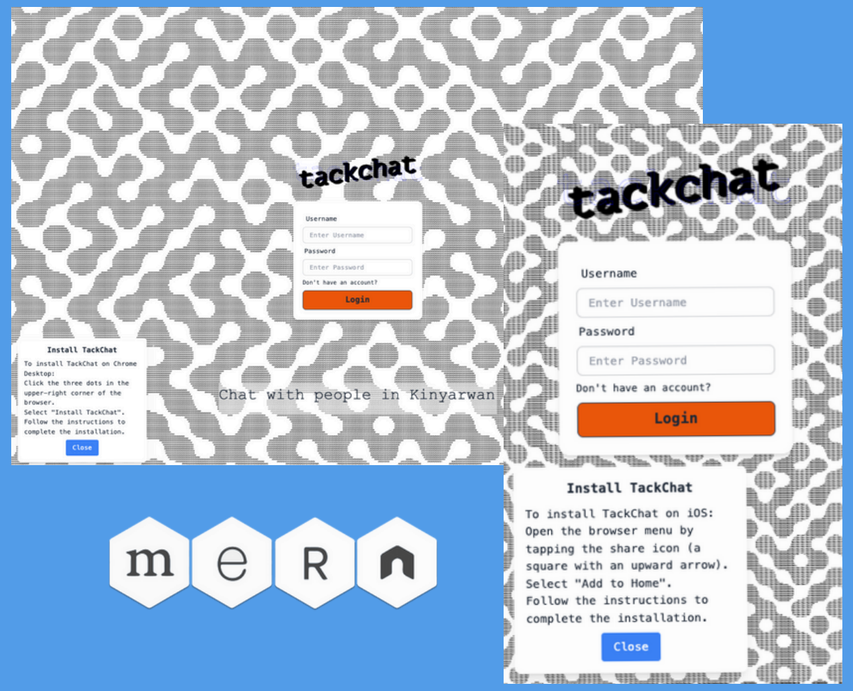
Home | **About Us** | App | Contact

L'APP

TackChat è un’applicazione web progettata per facilitare la comunicazione in tempo reale tra utenti. La peculiarità del progetto è l’abilità di tradurre automaticamente i messaggi tra lingue diverse, offrendo un’esperienza di comunicazione senza barriere linguistiche.

TackChat risponde a una sfida: la necessità di comunicare con persone di culture e lingue diverse senza dover ricorrere a traduttori esterni o manuali.

ARCHITETTURA



Frontend

- Costruito con React e ottimizzato con Vite

Backend

- Realizzato con Express per la logica delle API REST
- Utilizza MongoDB con Mongoose per la gestione dei modelli di dati

  - **API di Autenticazione (auth.routes.js)**
    - POST /api/auth/signup
    - POST /api/auth/login
    - POST /api/auth/logout
  - **API Utente (user.routes.js)**
    - GET /api/users/
    - GET /api/users/:id
  - **API Profilo (profile.routes.js)**
    - PUT /api/profile/updateProfile
  - **API Messaggi (message.routes.js)**
    - GET /api/messages/:id
    - POST /api/messages/send/:id

tackchat

FUNZIONALITÀ

1

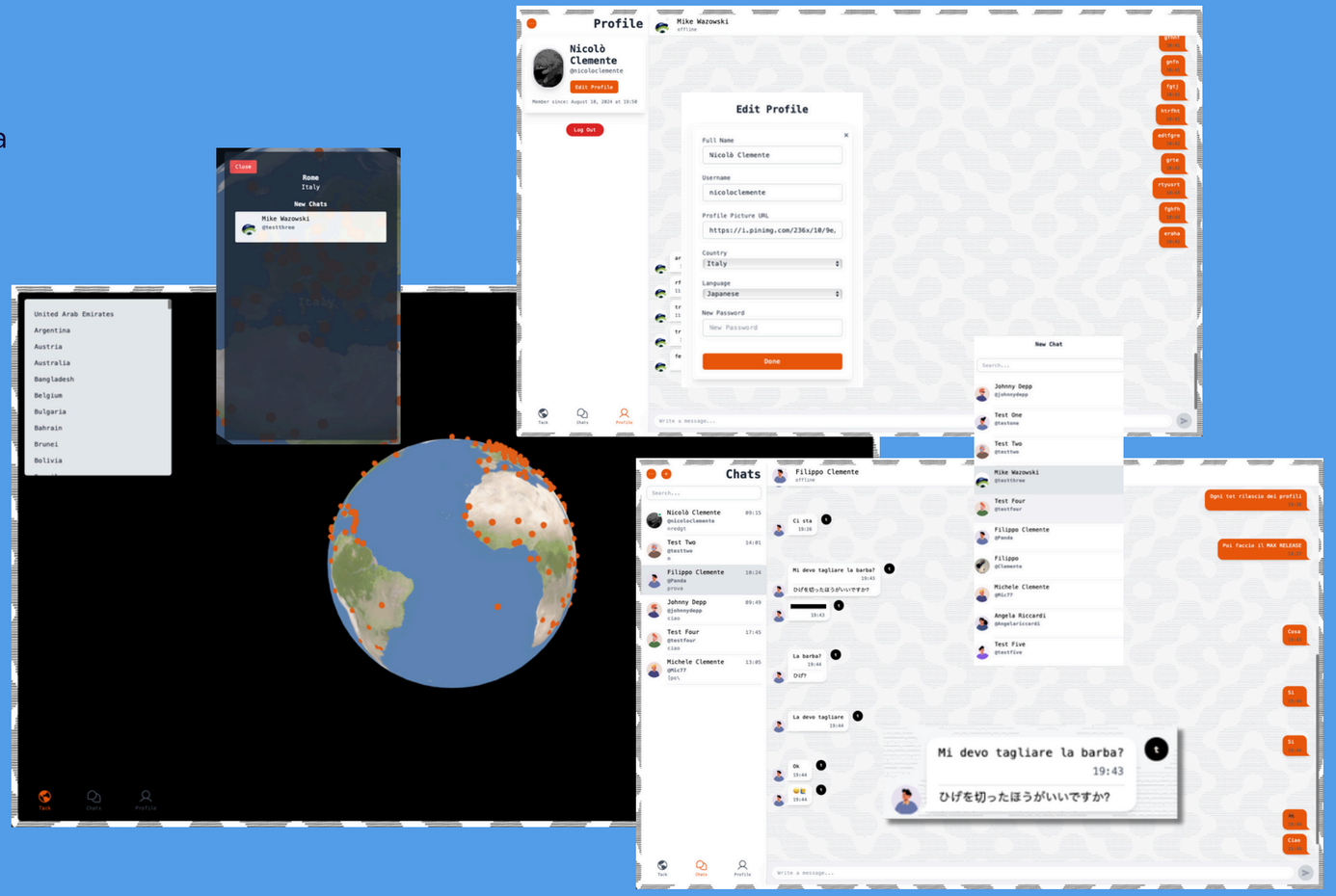
Tack
Esplora il globo e conosci nuovi utenti, senza la barriera della lingua

2

Chats
Tutte le tue conversazioni

3

Profile
Modifica e personalizza il tuo profilo



tackchat

Home About Us **App** Contact

FEATURES PRINCIPALI

Traduzione Automatica

google-cloud/translate traduce automaticamente i messaggi nella lingua desiderata dall’utente.

Real Time

Socket.io permette una comunicazione in tempo reale tra gli utenti. Gestisce la connessione costante tra client e server

Autenticazione Sicura

bcryptjs cripta le password, evitando che vengano salvate in chiaro. JWT garantisce sessioni sicure e mantiene il login.

PWA

permette agli utenti di installare l’app sul proprio dispositivo, offrendo un’esperienza rapida e leggera (33kb).

tackchat

ALTRE TECNOLOGIE UTILIZZATE

- **cookie-parser**: Gestisce i cookie per mantenere lo stato di sessione.
- **cors**: Consente comunicazione sicura tra frontend e backend su domini diversi.
- **dotenv**: Gestisce variabili d'ambiente, come chiavi API.
- **nodemon**: Riavvia automaticamente il server durante lo sviluppo a ogni modifica del codice.

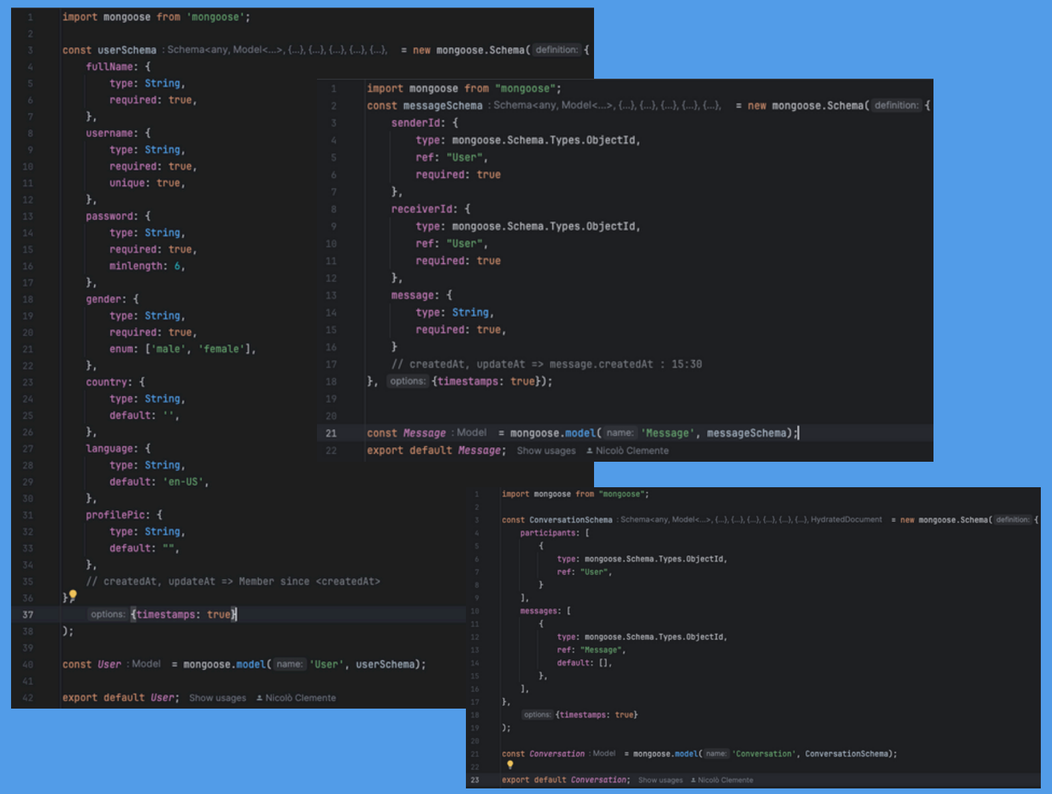
- **react-device-detect**: Ottimizza l'interfaccia utente a seconda del dispositivo per una migliore esperienza.
- **vite-plugin-pwa**: Trasforma l'app in una PWA, rendendola installabile e funzionante offline.

```
const isPhone = () => isMobileDevice && !isTablet;  Show usages

const Home = () => {  Show usages
    return isPhone() ? <HomeMobile /> : <HomeDesktop />;
};

export default Home;  Show usages
```

MODELLO DEI DATI



```
import mongoose from 'mongoose';

const userSchema : Schema<any, Model<...>, {...}, {...}, {...}, {...}, {...}, {...}> = new mongoose.Schema( definition: {
    fullName: {
        type: String,
        required: true,
    },
    username: {
        type: String,
        required: true,
        unique: true,
    },
    password: {
        type: String,
        required: true,
        minlength: 6,
    },
    gender: {
        type: String,
        required: true,
        enum: ['male', 'female'],
    },
    country: {
        type: String,
        default: '',
    },
    language: {
        type: String,
        default: 'en-US',
    },
    profilePic: {
        type: String,
        default: "",
    },
    // createdAt, updateAt => Member since <createdAt>
},
    options: {timestamps: true}
);

const User : Model = mongoose.model( name: 'User', userSchema);

export default User; Show usages  Nicolò Clemente
```

```
import mongoose from "mongoose";
const messageSchema : Schema<any, Model<...>, {...}, {...}, {...}, {...}, {...}, {...}> = new mongoose.Schema( definition: {
    senderId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "User",
        required: true
    },
    receiverId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: "User",
        required: true
    },
    message: {
        type: String,
        required: true,
    }
    // createdAt, updateAt => message.createdAt : 15:30
}, options: {timestamps: true});


const Message : Model = mongoose.model( name: 'Message', messageSchema);
export default Message; Show usages  Nicolò Clemente
```

```
import mongoose from "mongoose";

const ConversationSchema : Schema<any, Model<...>, {...}, {...}, {...}, {...}, {...}, HydratedDocument> = new mongoose.Schema( definition: {
    participants: [
        {
            type: mongoose.Schema.Types.ObjectId,
            ref: "User",
        }
    ],
    messages: [
        {
            type: mongoose.Schema.Types.ObjectId,
            ref: "Message",
            default: [],
        },
    ],
},
    options: {timestamps: true}
);

const Conversation : Model = mongoose.model( name: 'Conversation', ConversationSchema);

export default Conversation; Show usages  Nicolò Clemente
```

Il modello dei dati è centralizzato su MongoDB, che gestisce i dati relativi agli utenti, ai messaggi e alle conversazioni. Mongoose viene utilizzato per creare e gestire i modelli di dati in modo efficiente e coerente.

tackchat

Home About Us App Contact

INSTALLAZIONE

react-device-detect rileva le caratteristiche del dispositivo dell'utente e di conseguenza permette di gestire istruzioni ad-hoc per l'installazione, per ogni dispositivo.

```
<div>
    <p>To install TackChat on iOS:</p>
    <ul>
        <li>Open the browser menu by tapping the share icon (a square with an upward arrow).</li>
        <li>Select "Add to Home".</li>
        <li>Follow the instructions to complete the installation.</li>
    </ul>
</div>
```

```
<div>
    <p>To install TackChat on Android:</p>
    <ul>
        <li>Open the browser menu by tapping the three dots in the upper-right corner.</li>
        <li>Select "Add to Home Screen" or "Install".</li>
        <li>Follow the instructions to complete the installation.</li>
    </ul>
</div>
```

tackchat

Home About Us App Contact

GRAZIE PER L'ATTENZIONE

WWW.TACKCHAT.IT

POLITECNICO DI BARI