### **Documentazione degli Endpoint**

#### **/login (POST)**

* **Chiamata:**
  + Metodo: POST
  + Body:
    - email (string): Email dell'utente
    - password (string): Password dell'utente
* **Risposta:**
  + Successo (200):
    - Body:
      * token (string): Token JWT per l'autenticazione successiva
  + Errore:
    - Status 401: Credenziali non valide
    - Status 500: Errore interno del server

#### **/register (POST)**

* **Chiamata:**
  + Metodo: POST
  + Body:
    - email (string): Email dell'utente
    - password (string): Password dell'utente
* **Risposta:**
  + Successo (201):
    - Body:
      * message (string): Messaggio di conferma della creazione dell'utente
  + Errore:
    - Status 400: Email già registrata
    - Status 500: Errore interno del server

#### **/create-house (POST)**

* **Chiamata:**
  + Metodo: POST
  + Body:
    - image (string): URL dell'immagine dell'immobile
    - name (string): Nome dell'immobile
    - bathrooms (number): Numero di bagni
    - location (string): Località dell'immobile
    - superficie (number): Superficie dell'immobile
    - price (string): Prezzo dell'immobile
* **Autenticazione**: richiesta un token JWT valido nel header 'Authorization'
* **Risposta:**
  + Successo (201):
    - Body:
      * message (string): Messaggio di conferma della creazione dell'immobile
  + Errore:
    - Status 401: Token non fornito
    - Status 403: Token non valido
    - Status 500: Errore interno del server

#### **/houses (GET)**

* **Chiamata:**
  + Metodo: GET
* **Risposta:**
  + Successo (200):
    - Body:
      * Array di oggetti rappresentanti le case
  + Errore:
    - Status 500: Errore interno del server

#### **/houses/:id (PUT)**

* **Chiamata:**
  + Metodo: PUT
  + Params:
    - id (string): ID dell'immobile da aggiornare
  + Body:
    - image (string): URL dell'immagine dell'immobile
    - name (string): Nome dell'immobile
    - bathrooms (number): Numero di bagni
    - location (string): Località dell'immobile
    - superficie (number): Superficie dell'immobile
    - price (string): Prezzo dell'immobile
* **Autenticazione**: richiesta un token JWT valido nel header 'Authorization'
* **Risposta:**
  + Successo (200):
    - Body:
      * message (string): Messaggio di conferma dell'aggiornamento dell'immobile
  + Errore:
    - Status 401: Token non fornito
    - Status 403: Token non valido
    - Status 404: Immobile non trovato
    - Status 500: Errore interno del server

#### **/houses/:id (DELETE)**

* **Chiamata:**
  + Metodo: DELETE
  + Params:
    - id (string): ID dell'immobile da eliminare
* **Autenticazione**: richiesta un token JWT valido nel header 'Authorization'
* **Risposta:**
  + Successo (200):
    - Body:
      * message (string): Messaggio di conferma dell'eliminazione dell'immobile
  + Errore:
    - Status 401: Token non fornito
    - Status 403: Token non valido
    - Status 404: Immobile non trovato
    - Status 500: Errore interno del server

### **Utilizzo nel Frontend con React e Next.js**

* **Pagina: Home**
  + **Descrizione**: Pagina di accesso al gestionale.
  + **Endpoint Utilizzato**: /login (POST)
  + **Utilizzo**: Invia le credenziali (email e password) per l'autenticazione e ottiene un token JWT in risposta.
* **Pagina: Sito**
  + **Descrizione**: Pagina principale del sito che mostra gli immobili disponibili.
  + **Endpoint Utilizzato**: /houses (GET)
  + **Utilizzo**: Ottiene l'elenco degli immobili dal backend e li visualizza nella pagina.
* **Pagina: Rimuovi**
  + **Descrizione**: Pagina per rimuovere un immobile dal database.
  + **Endpoint Utilizzato**: /houses/:id (DELETE)
  + **Utilizzo**: Invia l'ID dell'immobile da eliminare al backend per rimuoverlo dal database.
* **Pagina: Modifica**
  + **Descrizione**: Pagina per modificare le informazioni di un immobile esistente nel database.
  + **Endpoint Utilizzato**: /houses/:id (PUT)
  + **Utilizzo**: Invia le nuove informazioni dell'immobile insieme all'ID dell'immobile da modificare al backend per aggiornare i dati nel database.
* **Pagina: Gestionale**
  + **Descrizione**: Pagina per aggiungere un nuovo immobile al database.
  + **Endpoint Utilizzato**: /create-house (POST)
  + **Utilizzo**: Invia le informazioni del nuovo immobile al backend per aggiungerlo al database.