### E-COMMERCE FRONT PAGE & BACK OFFICE

Il tuo obiettivo è di creare un finto e-commerce con le funzionalità di gestione dei prodotti.

Le features da implementare sono:

– Una frontpage, dove si vedono tutti i prodotti

– A backoffice da cui aggiungere nuovi prodotti

Nel backoffice, implementa un form per aggiungere un nuovo prodotto al database. Il prodotto deve essere strutturato così:

“`jsx

{

“\_id”: “5d318e1a8541744830bef139”, // SERVER GENERATED

“name”: “3310 cellphone”, // REQUIRED

“description”: “An unforgettable icon.”, // REQUIRED

“brand”: “Nokia”, // REQUIRED

“imageUrl”: “https://bit.ly/3CExjRa”, // REQUIRED

“price”: 100, // REQUIRED

“userId”: “admin”, // SERVER GENERATED

“createdAt”: “2021-09-19T09:32:10.535Z”, // SERVER GENERATED

“updatedAt”: “2021-09-19T09:32:10.535Z”, // SERVER GENERATED

“\_\_v”: 0 // SERVER GENERATED

}

“`

I campi che dicono “REQUIRED” sono obbligatori. Quelli “SERVER GENERATED”, invece, non serve che siano inviati all’API.

L’endpoint è:

`[https://striveschool-api.herokuapp.com/api/product/](https://striveschool-api.herokuapp.com/api/product/)`

Sia per GET che per POST.

🛑 \*\*IMPORTANTE\*\* 🛑

OGNI CHIAMATA DEVE ESSERE AUTENTICATE.

Ogni richiesta a questo API deve includere un token per ottenere l’accesso.

Puoi ottenere il token qui: [https://strive.school/studentlogin](<https://strive.school/studentlogin>)

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NDM2ZjIwMTMzYjE1MjAwMTQ3NjE3OTkiLCJpYXQiOjE2ODEzMjI0OTcsImV4cCI6MTY4MjUzMjA5N30.Ejen82rCmdq350yAqGo7sgtQ-XHKZy-Cj6yXAQ4kuYw

fetch("https://striveschool-api.herokuapp.com/api/put-your-endpoint-here/", {  
headers: {  
"Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NDM2ZjIwMTMzYjE1MjAwMTQ3NjE3OTkiLCJpYXQiOjE2ODEzMjI0OTcsImV4cCI6MTY4MjUzMjA5N30.Ejen82rCmdq350yAqGo7sgtQ-XHKZy-Cj6yXAQ4kuYw"  
}  
})

Esempio:

“`jsx

fetch(‘https://striveschool-api.herokuapp.com/api/product/’, {

headers: {

Authorization: ‘Bearer XXXXXXXXXXXXXXXXX’

}

})

“`

Dove “XXXXXXXXXXXXXXXXX“ deve essere sostituito dal token preso dalla pagina menzionata in precedenza.

Il token dura 14 giorni, se te ne serve uno nuovo, puoi ottenerlo con la seguente richiesta:

“`jsx

method: “POST”

endpoint: “https://striveschool-api.herokuapp.com/api/account/login”

request body:

{

“username”: // the username you registered with,

“password”: // the password you registered with

}

“`

Puoi iniziare con il template allegato.

### 🔴 CENTRO RISOLUZIONE PROBLEMI / FAQ:

Imparare a leggere gli errori è fondamentale\*\*! \*\*

\*\*Ricevo solo un array vuoto, perché?\*\*

L’API ti invierà solo i prodotti che TU hai aggiunti. Prova a creare qualcosa con POST e controlla di nuovo.

\*\*Ricevo un errore 500, come posso risolvere?\*\*

Controlla nella network tab del tuo inspector per vedere l’errore specifico.

Solitamente

1. Ti manca un field nel corpo

2. Hai una “duplicate key”, cioè il nome del prodotto esiste già

3. Hai inviato il tipo sbagliato di dati (una stringa invece di un numero o simili)

Questo esercizio ti chiede di creare un finto e-commerce con funzionalità di gestione dei prodotti.

Dovrai implementare una frontpage dove si vedono tutti i prodotti e un backoffice da cui aggiungere nuovi prodotti.

Nel backoffice, dovrai implementare un form per aggiungere un nuovo prodotto al database.

Il prodotto deve avere una struttura specifica con alcuni campi obbligatori e altri generati dal server.

L'endpoint per le richieste GET e POST è

`https://striveschool-api.herokuapp.com/api/product/`.

Ogni richiesta a questo API deve includere un token per ottenere l'accesso, che può essere ottenuto dal sito `https://strive.school/studentlogin`.

Il token dura 14 giorni e può essere ottenuto nuovamente con una richiesta POST all'endpoint

`https://striveschool-api.herokuapp.com/api/account/login` con il tuo nome utente e password registrati.

Per svolgere questo esercizio, potresti seguire questi passaggi:

1. Creare una frontpage per mostrare tutti i prodotti. Potresti utilizzare una richiesta GET all'endpoint `https://striveschool-api.herokuapp.com/api/product/` per ottenere la lista dei prodotti e visualizzarli nella pagina.

2. Creare un backoffice per gestire i prodotti. In questa sezione, potresti implementare un form per aggiungere un nuovo prodotto al database. Assicurati di includere tutti i campi obbligatori e di rispettare la struttura del prodotto specificata nell'esercizio.

3. Ottenere un token di accesso per effettuare le richieste all'API. Puoi ottenere il token dal sito `https://strive.school/studentlogin`. Ricorda di includere il token in ogni richiesta all'API utilizzando l'header `Authorization: Bearer <token>`.

4. Utilizzare una richiesta POST all'endpoint `https://striveschool-api.herokuapp.com/api/product/` per aggiungere un nuovo prodotto al database. Assicurati di includere il token di accesso nella richiesta.

5. Se il token scade dopo 14 giorni, puoi ottenerne uno nuovo effettuando una richiesta POST all'endpoint `https://striveschool-api.herokuapp.com/api/account/login` con il tuo nome utente e password registrati.

Ricorda di seguire attentamente le istruzioni dell'esercizio e di testare il tuo codice per assicurarti che tutto funzioni correttamente.

Sì, posso aiutarti a capire come svolgere l'esercizio punto per punto:

1. **\*\*Creare una frontpage per mostrare tutti i prodotti\*\*: Per creare la frontpage, potresti utilizzare una richiesta GET all'endpoint `https://striveschool-api.herokuapp.com/api/product/` per ottenere la lista dei prodotti. Una volta ottenuti i dati, potresti utilizzare il tuo framework o libreria preferita per creare una pagina web che mostri tutti i prodotti.**

Per creare una richiesta GET a un endpoint, puoi utilizzare la funzione `fetch()` di JavaScript. Ecco un esempio di come effettuare una richiesta GET all'endpoint `https://striveschool-api.herokuapp.com/api/product/`:

```javascript

fetch('https://striveschool-api.herokuapp.com/api/product/')

.then(response => response.json())

.then(data => {

// gestisci i dati ricevuti qui

console.log(data);

})

.catch(error => {

// gestisci eventuali errori qui

console.error(error);

});

```

In questo esempio, la funzione `fetch()` effettua una richiesta GET all'endpoint specificato e restituisce una promessa. Utilizziamo il metodo `.then()` per gestire la risposta e convertirla in un oggetto JSON utilizzando il metodo `response.json()`. Successivamente, utilizziamo un altro metodo `.then()` per gestire i dati ricevuti e visualizzarli nella console. Infine, utilizziamo il metodo `.catch()` per gestire eventuali errori che potrebbero verificarsi durante la richiesta.

Ricorda che, per effettuare richieste all'API specificata nell'esercizio, dovrai includere il token di accesso nell'header `Authorization` della richiesta.

1. **\*\*Creare un backoffice per gestire i prodotti\*\*: Per creare il backoffice, potresti creare una nuova pagina web con un form per aggiungere un nuovo prodotto al database. Assicurati di includere tutti i campi obbligatori specificati nell'esercizio e di rispettare la struttura del prodotto.**

Per creare il backoffice e il form per aggiungere un nuovo prodotto al database, potresti seguire questi passaggi:

1. Creare una nuova pagina web per il backoffice. Potresti utilizzare il tuo framework o libreria preferita per creare la pagina.

1. Aggiungere un form alla pagina del backoffice. Potresti utilizzare elementi HTML come `<form>`, `<input>` e `<label>` per creare il form. Assicurati di includere tutti i campi obbligatori specificati nell'esercizio, come `name`, `description`, `brand`, `imageUrl` e `price`.

Per creare un form HTML, puoi utilizzare l'elemento `<form>` e gli elementi `<input>`, `<label>` e `<button>` per creare i campi del form e il pulsante di invio. Ecco un esempio di come creare un semplice form HTML per aggiungere un nuovo prodotto:

```html

<form id="product-form">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="description">Description:</label>

<input type="text" id="description" name="description" required><br><br>

<label for="brand">Brand:</label>

<input type="text" id="brand" name="brand" required><br><br>

<label for="imageUrl">Image URL:</label>

<input type="url" id="imageUrl" name="imageUrl" required><br><br>

<label for="price">Price:</label>

<input type="number" id="price" name="price" required><br><br>

<button type="submit">Add Product</button>

</form>

```

In questo esempio, abbiamo creato un form con l'elemento `<form>` e abbiamo aggiunto diversi campi utilizzando gli elementi `<input>` e `<label>`. Abbiamo anche aggiunto un pulsante di invio utilizzando l'elemento `<button>`.

Ricorda di impostare l'attributo `name` per ogni elemento `<input>` in modo da poter raccogliere facilmente i dati inseriti dall'utente nel form. Inoltre, puoi utilizzare l'attributo `required` per rendere obbligatorio l'inserimento di un valore in un campo del form.

Una volta creato il form HTML, potrai utilizzare JavaScript per gestire l'evento di invio del form e raccogliere i dati inseriti dall'utente.

3. Aggiungere un pulsante per inviare il form. Potresti utilizzare un elemento `<button>` o `<input type="submit">` per creare il pulsante.

4. Aggiungere un gestore per l'evento di invio del form. Potresti utilizzare il metodo `addEventListener()` per aggiungere un gestore per l'evento `submit` del form. Nel gestore dell'evento, potresti raccogliere i dati inseriti dall'utente nel form e utilizzarli per creare un nuovo prodotto.

5. Effettuare una richiesta POST all'endpoint `https://striveschool-api.herokuapp.com/api/product/` per aggiungere il nuovo prodotto al database. Potresti utilizzare la funzione `fetch()` di JavaScript per effettuare la richiesta POST, includendo i dati del nuovo prodotto nel corpo della richiesta e il token di accesso nell'header `Authorization` della richiesta.

Per effettuare una richiesta POST all'endpoint `https://striveschool-api.herokuapp.com/api/product/` per aggiungere un nuovo prodotto al database utilizzando la funzione `fetch()` di JavaScript, potresti seguire questi passaggi:

1. Raccogliere i dati del nuovo prodotto dal form. Potresti utilizzare il metodo `querySelector()` per selezionare gli elementi `<input>` del form e il loro valore per creare un oggetto con i dati del nuovo prodotto.

```javascript

const productForm = document.querySelector('#product-form');

productForm.addEventListener('submit', event => {

event.preventDefault();

const newProduct = {

name: productForm.querySelector('#name').value,

description: productForm.querySelector('#description').value,

brand: productForm.querySelector('#brand').value,

imageUrl: productForm.querySelector('#imageUrl').value,

price: productForm.querySelector('#price').value

};

// ...

});

```

2. Creare le opzioni per la richiesta POST. Potresti creare un oggetto con le opzioni per la richiesta POST, includendo il metodo `POST`, i dati del nuovo prodotto nel corpo della richiesta e il token di accesso nell'header `Authorization` della richiesta.

```javascript

const options = {

method: 'POST',

headers: {

'Content-Type': 'application/json',

'Authorization': 'Bearer <token>'

},

body: JSON.stringify(newProduct)

};

```

3. Effettuare la richiesta POST utilizzando la funzione `fetch()`. Potresti utilizzare la funzione `fetch()` per effettuare la richiesta POST all'endpoint specificato, passando l'URL dell'endpoint e le opzioni create in precedenza come argomenti.

```javascript

fetch('https://striveschool-api.herokuapp.com/api/product/', options)

.then(response => response.json())

.then(data => {

// gestisci i dati ricevuti qui

console.log(data);

})

.catch(error => {

// gestisci eventuali errori qui

console.error(error);

});

```

In questo esempio, abbiamo utilizzato la funzione `fetch()` per effettuare una richiesta POST all'endpoint specificato, passando l'URL dell'endpoint e le opzioni create in precedenza come argomenti. Utilizziamo il metodo `.then()` per gestire la risposta e convertirla in un oggetto JSON utilizzando il metodo `response.json()`. Successivamente, utilizziamo un altro metodo `.then()` per gestire i dati ricevuti e visualizzarli nella console. Infine, utilizziamo il metodo `.catch()` per gestire eventuali errori che potrebbero verificarsi durante la richiesta.

Per creare un form HTML, puoi utilizzare l'elemento `<form>` e gli elementi `<input>`, `<label>` e `<button>` per creare i campi del form e il pulsante di invio. Ecco un esempio di come creare un semplice form HTML per aggiungere un nuovo prodotto:

```html

<form id="product-form">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required><br><br>

<label for="description">Description:</label>

<input type="text" id="description" name="description" required><br><br>

<label for="brand">Brand:</label>

<input type="text" id="brand" name="brand" required><br><br>

<label for="imageUrl">Image URL:</label>

<input type="url" id="imageUrl" name="imageUrl" required><br><br>

<label for="price">Price:</label>

<input type="number" id="price" name="price" required><br><br>

<button type="submit">Add Product</button>

</form>

```

In questo esempio, abbiamo creato un form con l'elemento `<form>` e abbiamo aggiunto diversi campi utilizzando gli elementi `<input>` e `<label>`. Abbiamo anche aggiunto un pulsante di invio utilizzando l'elemento `<button>`.

Ricorda di impostare l'attributo `name` per ogni elemento `<input>` in modo da poter raccogliere facilmente i dati inseriti dall'utente nel form. Inoltre, puoi utilizzare l'attributo `required` per rendere obbligatorio l'inserimento di un valore in un campo del form.

Una volta creato il form HTML, potrai utilizzare JavaScript per gestire l'evento di invio del form e raccogliere i dati inseriti dall'utente.

3. \*\*Ottenere un token di accesso per effettuare le richieste all'API\*\*: Per ottenere il token di accesso, puoi visitare il sito `https://strive.school/studentlogin` e seguire le istruzioni per ottenere il token. Una volta ottenuto il token, dovrai includerlo in ogni richiesta all'API utilizzando l'header `Authorization: Bearer <token>`.

4. \*\*Utilizzare una richiesta POST per aggiungere un nuovo prodotto al database\*\*: Per aggiungere un nuovo prodotto al database, potresti utilizzare una richiesta POST all'endpoint `https://striveschool-api.herokuapp.com/api/product/`. Assicurati di includere il token di accesso nella richiesta e di inviare i dati del nuovo prodotto nel corpo della richiesta.

5. \*\*Ottenere un nuovo token se quello attuale scade\*\*: Se il token scade dopo 14 giorni, puoi ottenerne uno nuovo effettuando una richiesta POST all'endpoint `https://striveschool-api.herokuapp.com/api/account/login` con il tuo nome utente e password registrati nel corpo della richiesta.

Spero che queste informazioni ti aiutino a capire come svolgere l'esercizio passo dopo passo.