

Task: Create an Express Server with Supabase Authentication

Goal:

Build a Node.js + Express server with **2 endpoints**:

1. `POST /login` — authenticates the user using a Supabase table.
 2. `POST /products` — returns product data from Supabase, **only if** the user is authenticated.
-

Setup Instructions:

1. Import the `classicmodels` database (you can use CSV) into your Supabase project.
 - You only need the `products` table.
2. Create a new Supabase table:
 - Table name: `users`
 - Columns: `username` (text), `password` (text)

Important Note:

In real apps, we **never store passwords like this**. We should always hash them.

For this exercise, plain text is fine — but this is **not secure** and just for learning.

Environment Variables (`.env` file)

You must use a `.env` file to store **all sensitive or changeable values**.

At a minimum, store:

- Supabase URL

- Supabase anonymous public key
- Your Express server port

You can also store any other values you want in the `.env`.

Make sure to load the values using the `dotenv` package: (or use with cli)

```
npm install dotenv
```

And at the top of your JS file:

```
import dotenv from 'dotenv';
dotenv.config();
```

Do **not** hard-code secrets or config values directly in your code.

What the Server Should Do:

1. POST /login

- Accepts `username` and `password` in the request **body** (JSON).
- Server checks in the Supabase `users` table:
 - Does this user exist?
 - Does the password match?
- Respond:
 - "Login successful" if OK.
 - "Wrong username or password" if not.

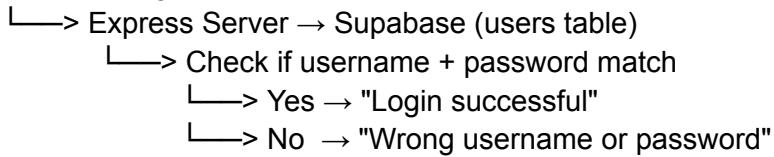
2. GET /products

- User sends the **same** `username` and `password` in the body or headers.
- Server:

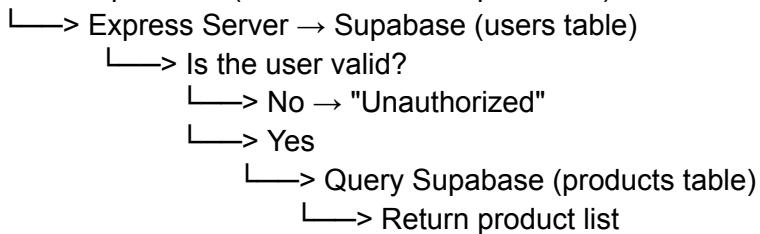
1. Checks if the credentials are correct in Supabase `users` table.
 2. If yes → query and return all rows from the `products` table.
 3. If no → respond with error: "`Unauthorized`".
-

Flow of Operations

User → POST /login



User → GET /products (send username + password)



What to Focus On:

- Organize your Express server with 2 endpoints
- Connect to Supabase using the official JS client
- Store config in `.env` and load with `dotenv/cli`
- Use clean error handling and clear messages
- Never send database credentials in your code