

Dokumentimi i Projektit me Supabase dhe Postman

Titulli i Projektit: Menaxhimi i Produkteve dhe Përdoruesve duke përdorur Supabase, Node.js dhe Postman

PËRMBLEDHJE E PROJEKTIT:

Ky projekt përfshin një aplikacion Node.js me Supabase si bazë të të dhënave. Realizon funksione për:

- Regjistrim dhe kyçje të përdoruesve
- Gjenerim token-i JWT
- Marrje profili me autentifikim
- CRUD për produkte me lidhje direkte në Supabase

STRUKTURA E PROJEKTIT

1. Përdorimi i Supabase si database

- U krijuan dy tabela: users dhe products
- Kolonat:
 - users: id (uuid), email, password
 - products: id (int8), name, price (float8) 

2. Regjistrimi dhe Kyçja (Auth) me JWT

- U krijuar register dhe login në usersController.js
- Gjenerimi i tokenit pas kyçjes:

```
const token = generateToken({ id: user.id, email: user.email });
```

- JWT ruan të dhëna në req.user pas dekodimit në middleware.  *(login POST), Screenshot #4 (token)*

3. Autentikimi me Middleware

- Krijuam një middleware authMiddleware.js që kontrollon tokenin:

```
const decoded = jwt.verify(token, secret);
```

- Nëse tokeni është valid, shkon në next().  Shih: Screenshot #5 – përdorimi në GET /users/me
-

4. Profili i përdoruesit (GET /users/me)

- Ky endpoint i shfaq të dhënat e përdoruesit që është autentifikuar:

```
res.json({ user: req.user });
```

- Funksionon vetëm nëse dërgohet token-i në Authorization header.
-

5. Produktet – CRUD në Supabase

U ndërtuan 4 funksione për productsController.js:

- **GET /products** – Merr të gjitha produktet nga Supabase
 - **POST /products** – Shton produkt të ri
 - **PUT /products/:id** – Përditëson një produkt
 - **DELETE /products/:id** – Fshin një produkt
-

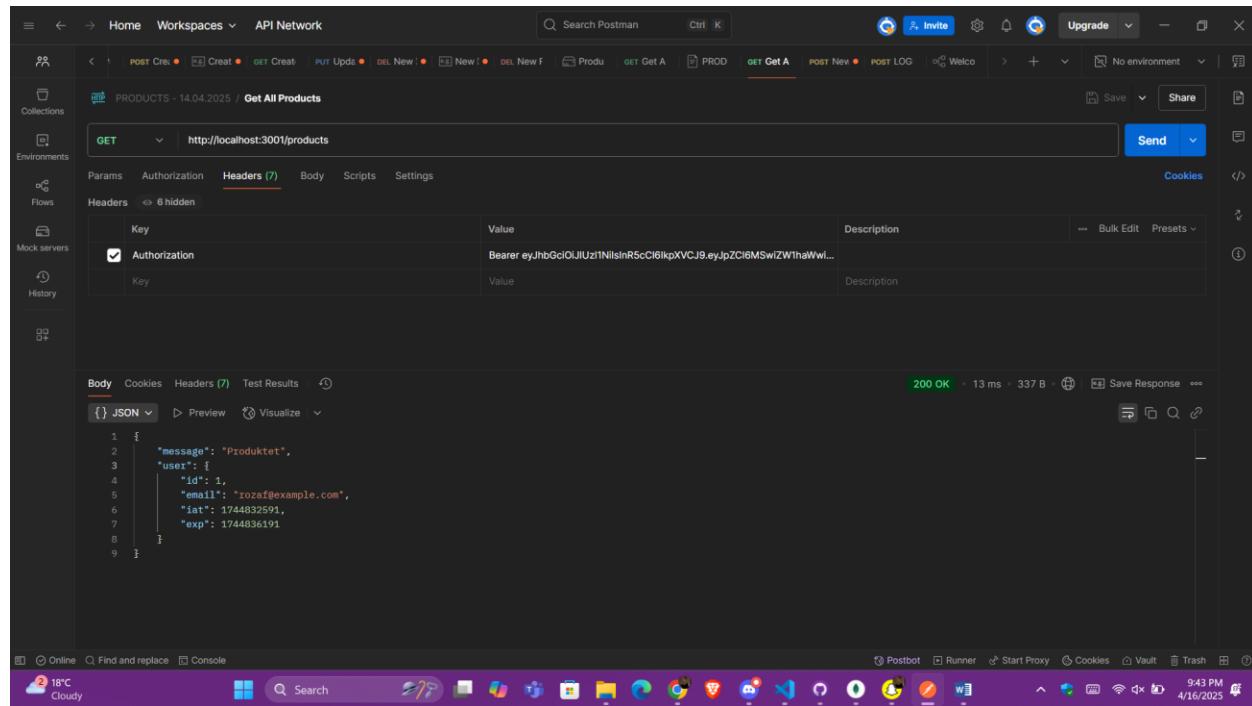
TESTIMI ME POSTMAN

- Të gjitha request-et u testuan në Postman
 - U dërguan Authorization: Bearer <token> për endpoint-et e mbrojtura
-

FUNKSIONALITETI FINAL

- CRUD për produkte funksionon me autentifikim dhe ruajtje në Supabase
- Autentifikimi me email + password, gjeneron token JWT
- Endpoint /users/me tregon profilin

1. Në këtë hap kam përdorur endpoint-in `/users/me` për të testuar nëse token-i i autentikimit funksionon. Siç shihet në përgjigje, token-i është i vlefshëm dhe kthen të dhënrat e përdoruesit të kyçur, përfshirë `id`-në dhe `email`-in. Kjo dëshmon që middleware për verifikimin e JWT është integruar saktë."



The screenshot shows the Postman application interface. A collection named "PRODUCTS - 14.04.2025 / Get All Products" is selected. A GET request is made to `http://localhost:3001/products`. In the Headers tab, there is an Authorization header set to `Bearer eyJhbGciOiJIUzI1Nl...
18°C Cloudy`. The response status is 200 OK, and the response body is displayed in JSON format:

```
1 {  
2   "message": "Produktet",  
3   "user": {  
4     "id": 1,  
5     "email": "roza@example.com",  
6     "iat": 1744832591,  
7     "exp": 1744836191  
8   }  
9 }
```

2. Në këtë hap, kam testuar funksionalitetin e krijimit të produkteve përmes API-së. Me këtë kërkesë POST, unë kam dërguar të dhënat e produktit të ri (emri dhe çmimi), dhe si përgjigje kam marrë mesazhin 'Produkti u shtua'. Kjo tregon që endpoint-i për shtim të produkteve është implementuar dhe funksionon siç duhet. Të dhënat ruhen automatikisht në Supabase.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Workspaces', 'More', search, settings, and an 'Upgrade' button. Below the header, a toolbar has icons for creating new requests, deleting, and saving environments. The main workspace shows a 'PRODUCTS - 14.04.2025 / 2nd request' entry. The request type is 'POST' and the URL is 'http://localhost:3001/products?'. The 'Body' tab is selected, showing a JSON payload:

```
1 {
2   "name": "Kafe Express",
3   "price": 2.5
4 }
5
```

Below the request details, the response is displayed. It shows a '200 OK' status with a response time of 5 ms and a size of 264 B. The response body is:

```
{ } JSON ▾ ▶ Preview ⚡ Visualize ▾
```

```
1 {
2   "message": "Produkt u shtua"
3 }
```

At the bottom, there are tabs for 'Console' and various Postman tools like 'Postbot', 'Runner', 'Vault', and help icons.

3. Kjo është kërkesa për përditësim të një produkti. Në këtë rast, kam përdorur metodën `PUT` për të përditësuar produktin me `id`: 1. Produkti është përditësuar nga 'Kafe Express' në 'Kafe Filter' dhe çmimi nga 2.5 në 3.5. Kjo tregon që API-ja jonë mbështet përditësimë dhe i reflekton ato menjëherë në Supabase.

The screenshot shows the Postman application interface. At the top, there are tabs for 'Workspaces' and 'More'. Below the tabs, a navigation bar includes 'New' (red), 'POST LOG', 'Welco', 'POST 2nd', 'PUT 3rd' (orange), and 'No environment'. The main area shows a 'PUT' request to 'http://localhost:3001/products/1'. The 'Body' tab is selected, showing the JSON payload:

```
1 {  
2   "name": "Kafe Filter",  
3   "price": 3.5  
4 }  
5
```

The response section shows a green '200 OK' status with a response time of '33 ms' and a size of '324 B'. The response body is displayed as:

```
{ } JSON ▾ ▷ Preview ⚡ Visualize ▾
```

```
1 {  
2   "message": "Produkti u përditësua",  
3   "product": {  
4     "id": 1,  
5     "name": "Kafe Filter",  
6     "price": 3.5  
7   }  
8 }
```

At the bottom, there are icons for 'Console', 'Postbot', 'Runner', 'Vault', and help.

4. Këtu kam përdorur metodën `DELETE` për të fshirë një produkt nga databaza. Kam specifikuar id: 1 për produktin që dua të fshij. Siç shihet, produkti me emrin 'Kafe Filter' është fshirë me sukses. Kërkesa është mbrojtur me autentikim dhe pranohet vetëm nëse përdoruesi është i kyçur me token valid.

The screenshot shows the Postman interface with the following details:

- Request Method:** `DELETE`
- URL:** `http://localhost:3001/products/1`
- Headers:**
 - `Authorization`: `Bearer eyJhbGciOiJIUzI1NilsIn...`
- Response Status:** `200 OK`
- Response Body (JSON):**

```
1 {  
2   "message": "Produkti u fshi",  
3   "product": {  
4     "id": 1,  
5     "name": "Kafe Filter",  
6     "price": 3.5  
7   }  
8 }
```

5. Pas fshirjes së produktit nëpërmjet Postman, kam kontrolluar tabelën `products` në Supabase për të konfirmuar që fshirja ka ndodhur me sukses. Siç mund të shihet në këtë pamje, tabela është bosh, që tregon se veprimi ka qenë i suksesshëm.

The screenshot shows the Supabase Table Editor interface. At the top, there are several tabs: 'KUJDESI YI - Përkthim mesazhe' (highlighted), 'Detyra Postman Supabase', 'projekti-me-users-products-14', and 'REST API | Supabase Docs'. The main area is titled 'Table Editor' and shows the 'public' schema. A table named 'products' is selected, containing two columns: 'id' (int8) and 'name' (text). The table is currently empty, with a message stating 'This table is empty' and 'Add rows to your table to get started.' There is a button to 'Import data from CSV'. On the left sidebar, there are icons for creating new tables, managing schemas, and other database operations. The bottom of the screen shows a Windows taskbar with various application icons and system status indicators.

6. Pas fshirjes së produktit nëpërmjet Postman, kam kontrolluar tabelën `products` në Supabase për të konfirmuar që fshirja ka ndodhur me sukses. Siç mund të shihet në këtë pamje, tabela është bosh, që tregon se veprimi ka qenë i suksesshëm.

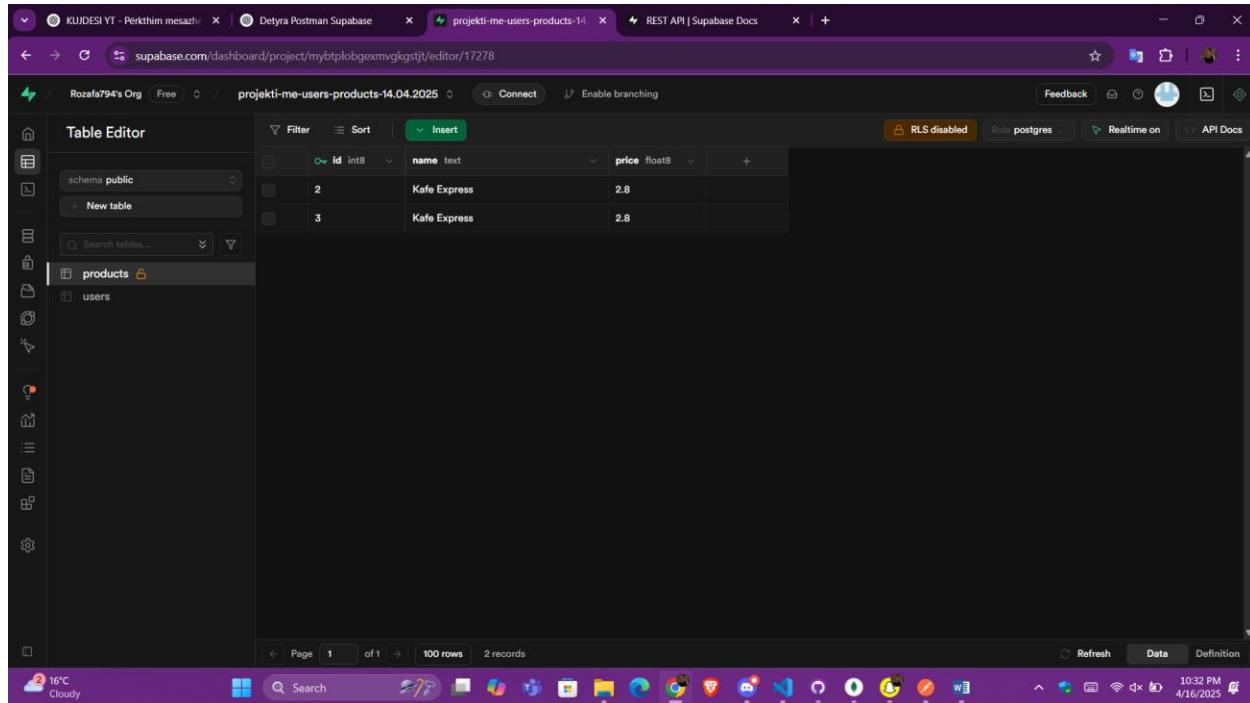
The screenshot shows the Postman interface with the following details:

- Header Bar:** Workspaces, More, Upgrade, Share, Save, No environment.
- Request Section:** Method: POST, URL: `http://localhost:3001/products`, Send button.
- Body Tab:** Params, Auth, Headers (9), Body (selected), Scripts, Settings. Body type: JSON. Content:

```
1 {  
2   "name": "Kafe Express",  
3   "price": 2.8  
4 }
```
- Response Section:** Status: 200 OK, Time: 500 ms, Size: 318 B. Response body:

```
1 {  
2   "message": "Produkti u shtua",  
3   "product": {  
4     "id": 3,  
5     "name": "Kafe Express",  
6     "price": 2.8  
7   }  
8 }
```
- Bottom Navigation:** Console, Postbot, Runner, Vault, Help.

7. Pas krijimit të produktit në Postman përmes një kërkese POST, e kam kontrolluar në Supabase nëse është regjistruar me sukses. Siç shihet në këtë pamje, të dhënat janë futur drejt në kolonat përkatëse të tabelës products.



The screenshot shows the Supabase Table Editor interface. On the left, there's a sidebar with icons for home, database, schema, table creation, search, and table selection. The table selected is 'products'. The main area displays a table with three columns: 'id', 'name', and 'price'. There are two rows of data: one with id 2, name 'Kafe Express', and price 2.8; and another with id 3, name 'Kafe Express', and price 2.8. At the bottom, there are buttons for 'Page', '1 of 1', '100 rows', and '2 records'. The status bar at the bottom right shows the date and time: 4/16/2025, 10:32 PM.

	id	name	price
2	Kafe Express	2.8	
3	Kafe Express	2.8	

8. Kjo pamje tregon që pasi përdoruesi `rozaf@example.com` u regjistrua në Supabase, arrita të realizoj login me sukses. Serveri ma ktheu një token autenticimi, që më pas e përdora për të qasur të dhëna të mbrojtura si profili dhe produktet.

The screenshot shows the Postman application interface. At the top, there's a header with 'Workspaces' and 'More' dropdowns, a search bar, and an 'Upgrade' button. Below the header, the URL is set to `http://localhost:3000/auth/login`. The method is set to 'POST'. The 'Body' tab is selected, showing a JSON payload:

```
1 {  
2   "email": "rozaf@example.com",  
3   "password": "sekret123"  
4 }
```

Below the body, the response status is '200 OK' with a duration of '429 ms' and a size of '501 B'. The response body is displayed in JSON format:

```
1 {  
2   "message": "Kyqja me sukses",  
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJpZCI6IjNhZDdiNDM3LTVkNDYtNDc20S1hY2ZhLTI0MGMxNjlhYjE5MCIsImVtYWlsIj  
oicm96YWZAZXhhbXBsZS5jb20iLCJpYXQiOjE3NDQ4MzcxODUsImV4cCI6MTc0NDg0MDc4  
NX0.zi7FBtVwNphq0kawXwo1Zb71PVTJ_39s3b2pPSLHz_w"
```

At the bottom, there are navigation icons for 'Console', 'Postbot', 'Runner', 'Vault', and help.

9. Pasi mora token nga login-i, e përdora në këtë kërkesë për me verifikon userin e kyçur. Kjo pamje tregon që token-i u pranu dhe më kthyen të dhënat e sakta të përdoruesit, gjë që dëshmon se autentikimi po funksionon.

The screenshot shows the Postman application interface. At the top, there are tabs for Workspaces, More, and several recent requests. The current request is a GET to `http://localhost:3000/users/me`. The Headers tab is selected, showing an `Authorization` header with the value `Bearer eyJhbGciOiJIUzI1Ni...` . The Body tab shows a JSON response with the following content:

```

1  {
2    "user": {
3      "id": "3ad7b437-5d46-4769-acfa-240c169ab190",
4      "email": "rozaf@example.com",
5      "iat": 1744837185,
6      "exp": 1744840785
7    }
8  }

```

The status bar at the bottom indicates a 200 OK response with a duration of 6 ms and a size of 352 B. The bottom navigation bar includes icons for Postbot, Runner, Vault, and various system status indicators like battery and signal strength.

10. Ky është tabela `users` në Supabase. Pasi bëra kërkesën POST për regjistrim, kjo tabelë u mbush me të dhënat e përdoruesit `rozaf@example.com`. `id` u gjenerua automatikisht si `uuid` nga Supabase, dhe kjo më ndihmoi ta identifikoj përdoruesin edhe gjatë gjenerimit të tokenit JWT.

The screenshot shows the Supabase Table Editor interface. The top navigation bar includes tabs for "KUJDESI YI - Përkthim mesazhe", "Detyra Postman Supabase", "projekti-me-users-products-14", "REST API | Supabase Docs", and a search bar for "supabase.com/dashboard/project/myblobgexmvqkgstjt/editor/17267". Below the navigation is a toolbar with "Feedback", "Google Lens", and other icons. The main area is titled "Table Editor" and shows the "public" schema. A table named "users" is selected, containing three columns: "id" (type: `uuid`), "email" (type: `text`), and "password" (type: `text`). There is one record listed: `3ad7b437-5d46-4769-acfa-240c169ab190`, `rozaf@example.com`, and `sekrett23`. The bottom of the screen shows a taskbar with various application icons and system status indicators like weather (16°C, Cloudy), date (4/16/2025), and time (11:02 PM).

	<code>id</code> <code>uuid</code>	<code>email</code> <code>text</code>	<code>password</code> <code>text</code>
	<code>3ad7b437-5d46-4769-acfa-240c169ab190</code>	<code>rozaf@example.com</code>	<code>sekrett23</code>