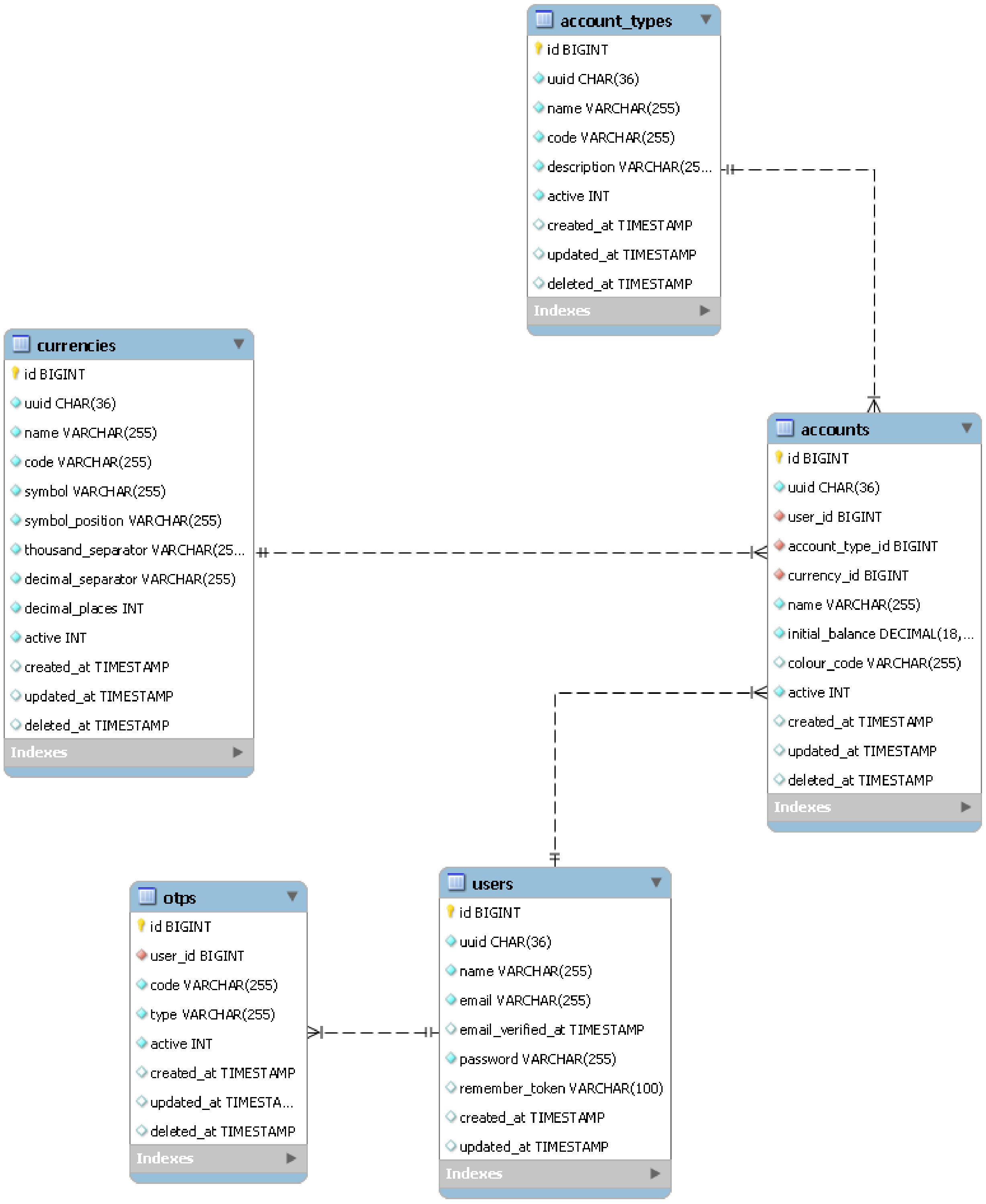
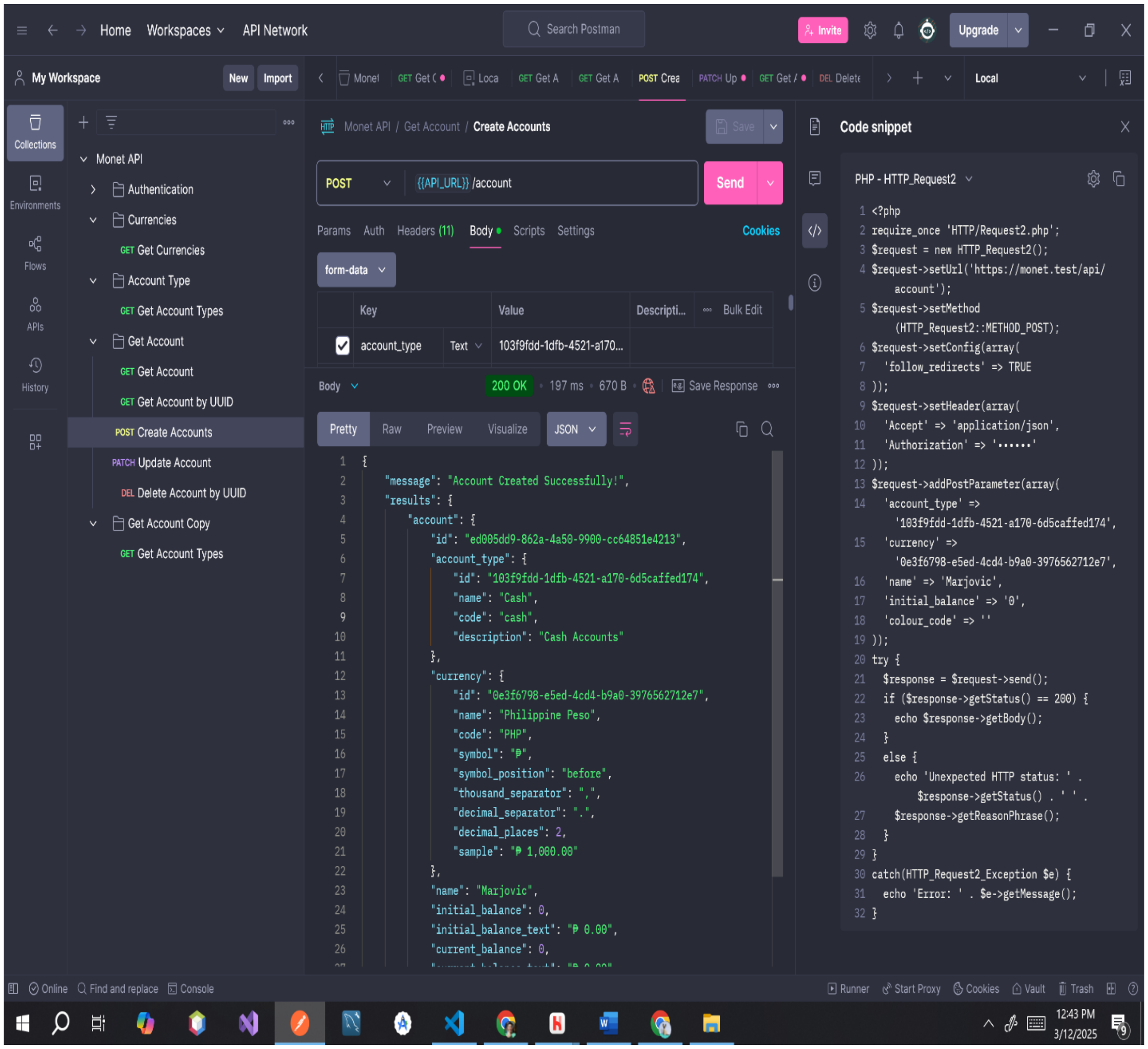
CRUD UI FRAMEWORK – we were working for it

DATABASE SCHEMA DIAGRAM



CREATE ACCOUNT



POST {{API_URL}} /account

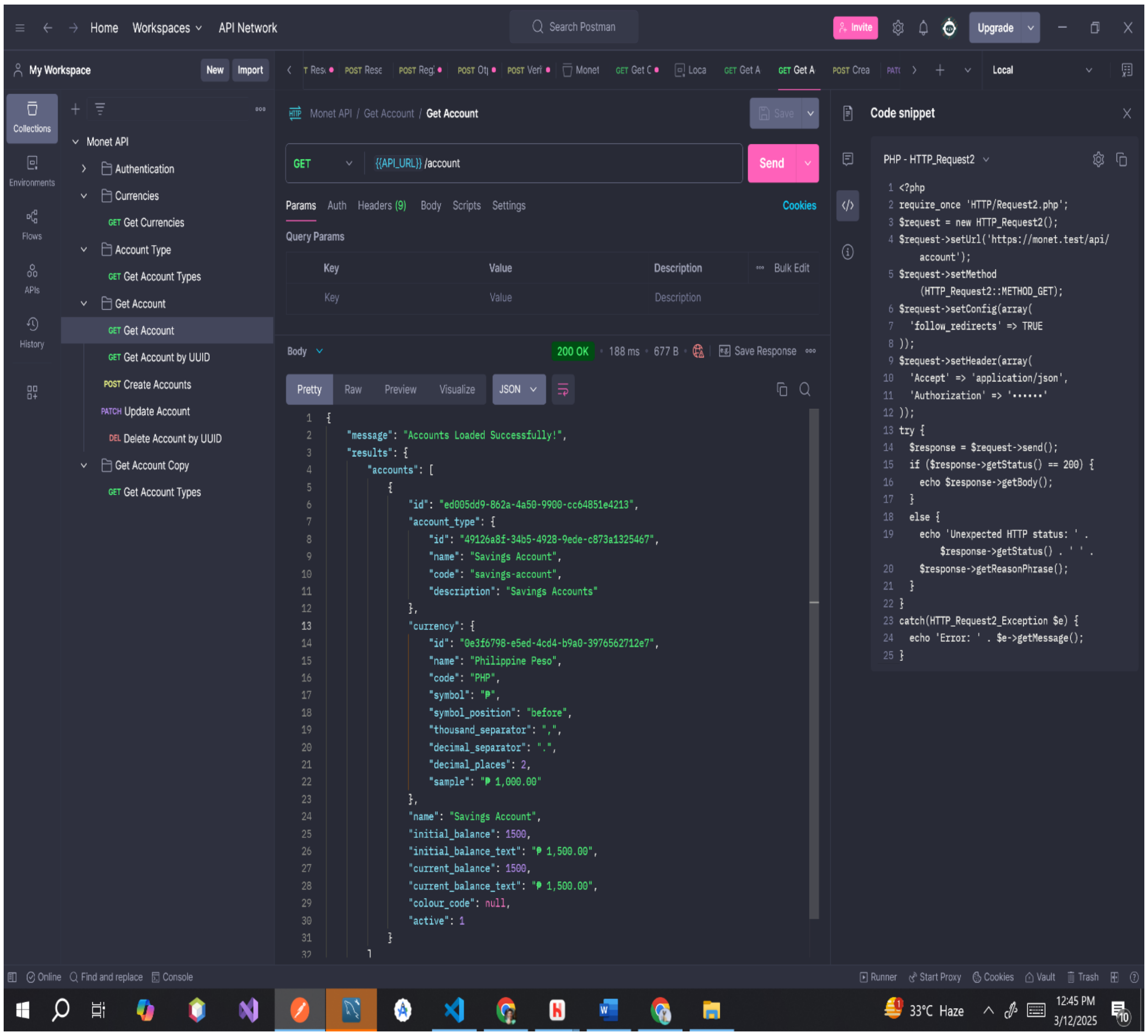
Key		Value
account_type	Text	103f9fdd-1dfb-4521-a170...

200 OK • 197 ms • 670 B

```json
{
    "message": "Account Created Successfully!",
    "results": {
        "account": {
            "id": "ed005dd9-862a-4a50-9900-cc64851e4213",
            "account_type": {
                "id": "103f9fdd-1dfb-4521-a170-6d5caffed174",
                "name": "Cash",
                "code": "cash",
                "description": "Cash Accounts"
            },
            "currency": {
                "id": "0e3f6798-e5ed-4cd4-b9a0-3976562712e7",
                "name": "Philippine Peso",
                "code": "PHP",
                "symbol": "₱",
                "symbol_position": "before",
                "thousand_separator": ",",
                "decimal_separator": ".",
                "decimal_places": 2,
                "sample": "₱ 1,000.00"
            },
            "name": "Marjovic",
            "initial_balance": 0,
            "initial_balance_text": "₱ 0.00",
            "current_balance": 0,
```

```php
<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('https://monet.test/api/
    account');
$request->setMethod
    (HTTP_Request2::METHOD_POST);
$request->setConfig(array(
  'follow_redirects' => TRUE
));
$request->setHeader(array(
  'Accept' => 'application/json',
  'Authorization' => '••••••'
));
$request->addPostParameter(array(
  'account_type' =>
    '103f9fdd-1dfb-4521-a170-6d5caffed174',
  'currency' =>
    '0e3f6798-e5ed-4cd4-b9a0-3976562712e7',
  'name' => 'Marjovic',
  'initial_balance' => '0',
  'colour_code' => ''
));
try {
  $response = $request->send();
  if ($response->getStatus() == 200) {
    echo $response->getBody();
  }
  else {
    echo 'Unexpected HTTP status: ' .
        $response->getStatus() . ' ' .
    $response->getReasonPhrase();
  }
}
catch(HTTP_Request2_Exception $e) {
  echo 'Error: ' . $e->getMessage();
}
```

READ ACCOUNT



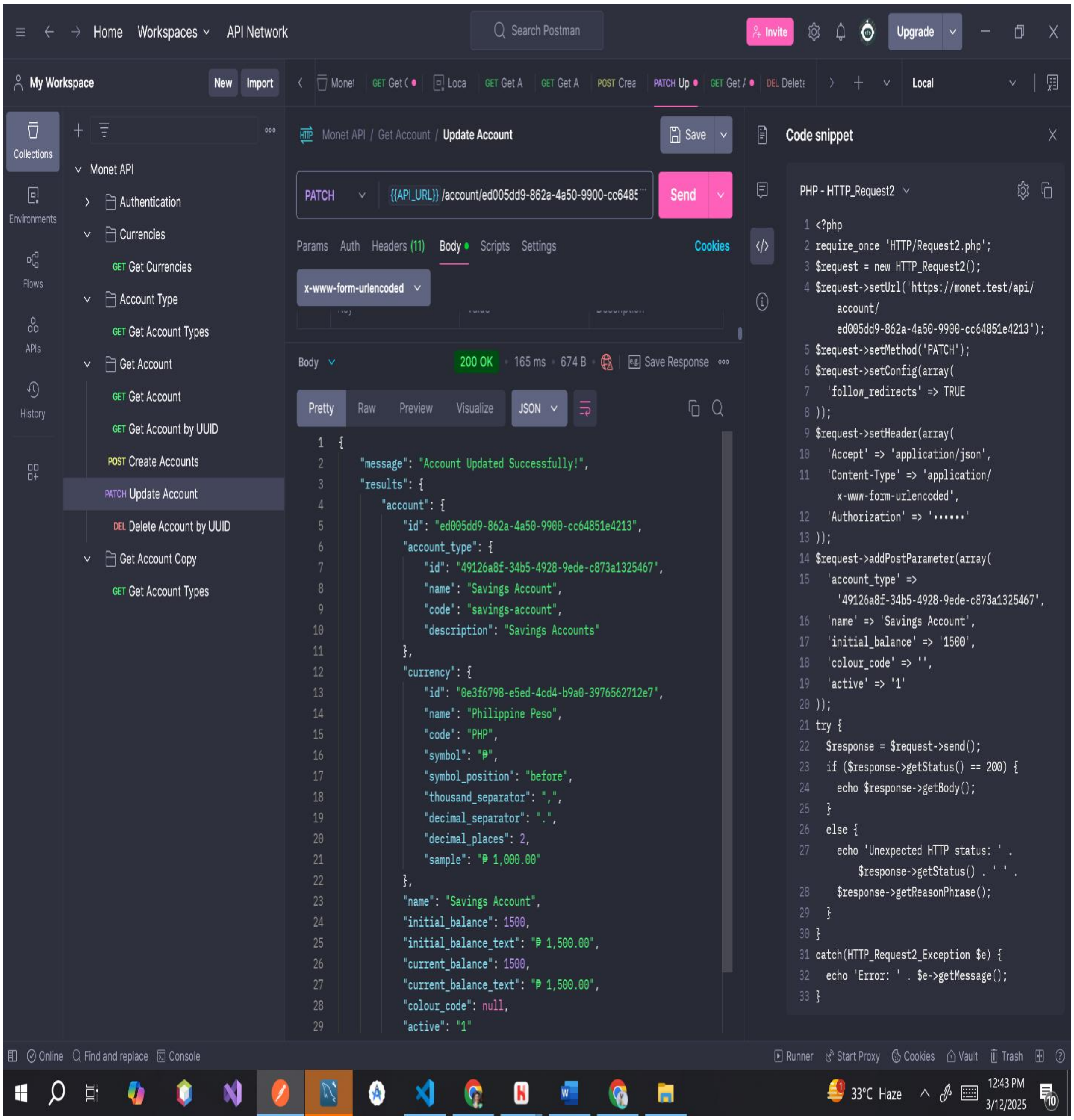
GET {{API_URL}} /account

200 OK · 188 ms · 677 B

```
{
    "message": "Accounts Loaded Successfully!",
    "results": {
        "accounts": [
            {
                "id": "ed005dd9-862a-4a50-9900-cc64851e4213",
                "account_type": {
                    "id": "49126a8f-34b5-4928-9ede-c873a1325467",
                    "name": "Savings Account",
                    "code": "savings-account",
                    "description": "Savings Accounts"
                },
                "currency": {
                    "id": "0e3f6798-e5ed-4cd4-b9a0-3976562712e7",
                    "name": "Philippine Peso",
                    "code": "PHP",
                    "symbol": "₱",
                    "symbol_position": "before",
                    "thousand_separator": ",",
                    "decimal_separator": ".",
                    "decimal_places": 2,
                    "sample": "₱ 1,000.00"
                },
                "name": "Savings Account",
                "initial_balance": 1500,
                "initial_balance_text": "₱ 1,500.00",
                "current_balance": 1500,
                "current_balance_text": "₱ 1,500.00",
                "colour_code": null,
                "active": 1
            }
        ]
```

```php
<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('https://monet.test/api/
    account');
$request->setMethod
    (HTTP_Request2::METHOD_GET);
$request->setConfig(array(
  'follow_redirects' => TRUE
));
$request->setHeader(array(
  'Accept' => 'application/json',
  'Authorization' => '••••••'
));
try {
  $response = $request->send();
  if ($response->getStatus() == 200) {
    echo $response->getBody();
  }
  else {
    echo 'Unexpected HTTP status: ' .
        $response->getStatus() . ' ' .
    $response->getReasonPhrase();
  }
}
catch(HTTP_Request2_Exception $e) {
  echo 'Error: ' . $e->getMessage();
}
```

UPDATE ACCOUNT



PATCH {{API_URL}} /account/ed005dd9-862a-4a50-9900-cc6485

Body: x-www-form-urlencoded

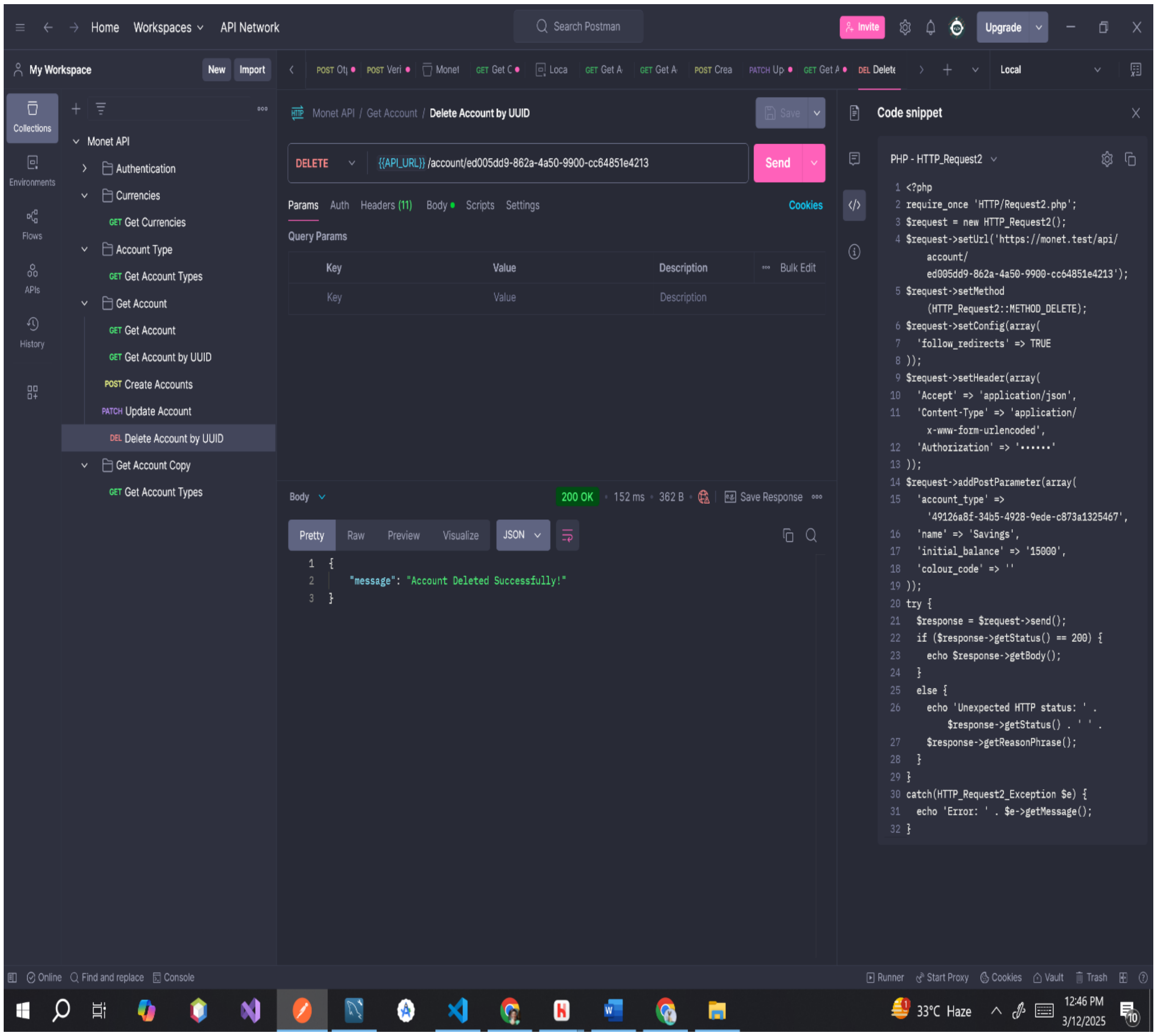
200 OK · 165 ms · 674 B

```
{
    "message": "Account Updated Successfully!",
    "results": {
        "account": {
            "id": "ed005dd9-862a-4a50-9900-cc64851e4213",
            "account_type": {
                "id": "49126a8f-34b5-4928-9ede-c873a1325467",
                "name": "Savings Account",
                "code": "savings-account",
                "description": "Savings Accounts"
            },
            "currency": {
                "id": "0e3f6798-e5ed-4cd4-b9a0-3976562712e7",
                "name": "Philippine Peso",
                "code": "PHP",
                "symbol": "₱",
                "symbol_position": "before",
                "thousand_separator": ",",
                "decimal_separator": ".",
                "decimal_places": 2,
                "sample": "₱ 1,000.00"
            },
            "name": "Savings Account",
            "initial_balance": 1500,
            "initial_balance_text": "₱ 1,500.00",
            "current_balance": 1500,
            "current_balance_text": "₱ 1,500.00",
            "colour_code": null,
            "active": "1"
```

PHP - HTTP_Request2

```php
<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('https://monet.test/api/account/ed005dd9-862a-4a50-9900-cc64851e4213');
$request->setMethod('PATCH');
$request->setConfig(array(
  'follow_redirects' => TRUE
));
$request->setHeader(array(
  'Accept' => 'application/json',
  'Content-Type' => 'application/x-www-form-urlencoded',
  'Authorization' => '••••••'
));
$request->addPostParameter(array(
  'account_type' => '49126a8f-34b5-4928-9ede-c873a1325467',
  'name' => 'Savings Account',
  'initial_balance' => '1500',
  'colour_code' => '',
  'active' => '1'
));
try {
  $response = $request->send();
  if ($response->getStatus() == 200) {
    echo $response->getBody();
  }
  else {
    echo 'Unexpected HTTP status: ' . $response->getStatus() . ' ' .
    $response->getReasonPhrase();
  }
}
catch(HTTP_Request2_Exception $e) {
  echo 'Error: ' . $e->getMessage();
}
```

DELETE ACCOUNT



Monet API / Get Account / Delete Account by UUID

DELETE {{API_URL}} /account/ed005dd9-862a-4a50-9900-cc64851e4213

Params Auth Headers (11) Body Scripts Settings

Query Params

Key	Value	Description
Key	Value	Description

Body 200 OK · 152 ms · 362 B

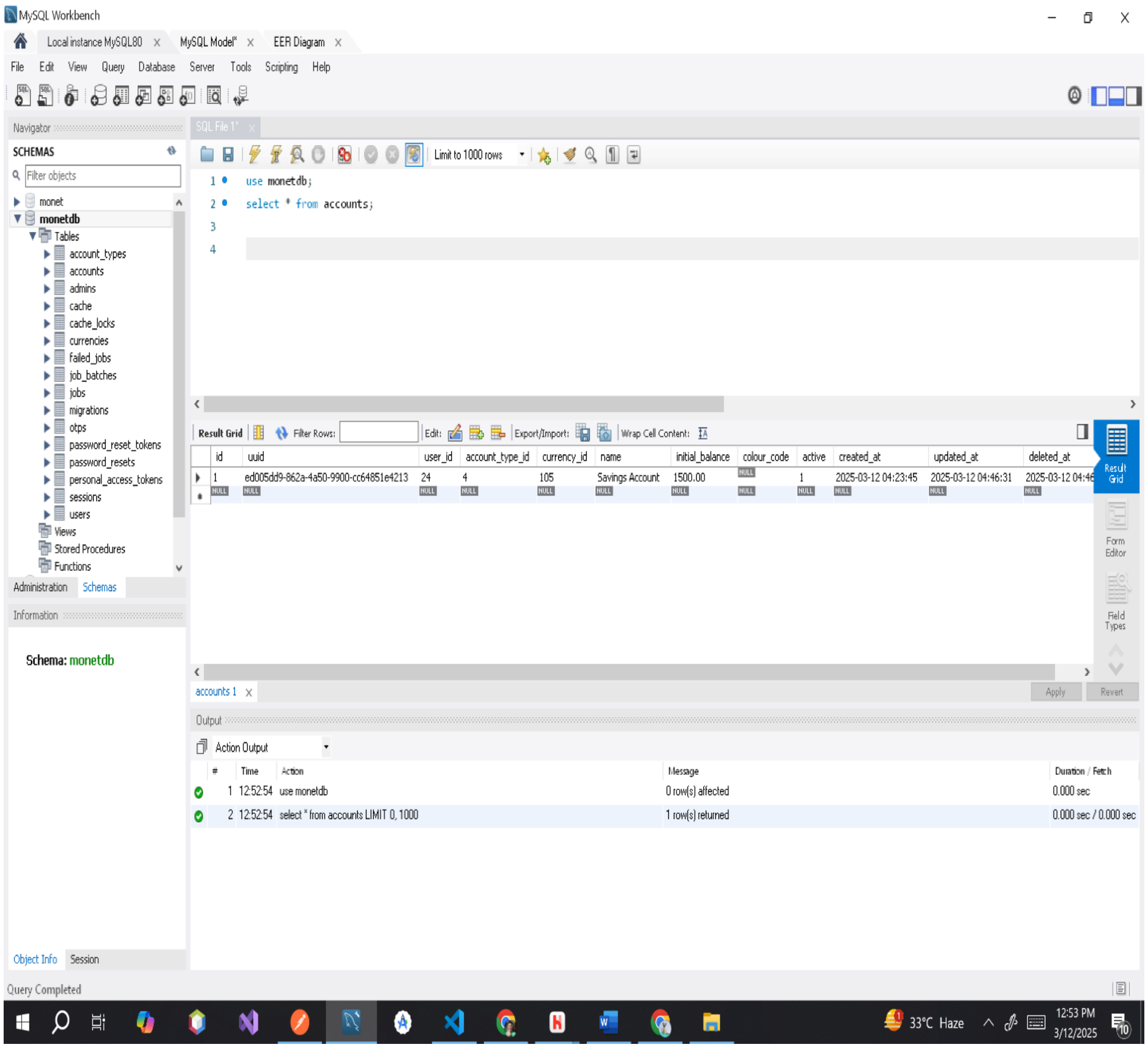
```
{
    "message": "Account Deleted Successfully!"
}
```

Code snippet

PHP - HTTP_Request2

```
<?php
require_once 'HTTP/Request2.php';
$request = new HTTP_Request2();
$request->setUrl('https://monet.test/api/
    account/
    ed005dd9-862a-4a50-9900-cc64851e4213');
$request->setMethod
    (HTTP_Request2::METHOD_DELETE);
$request->setConfig(array(
  'follow_redirects' => TRUE
));
$request->setHeader(array(
  'Accept' => 'application/json',
  'Content-Type' => 'application/
    x-www-form-urlencoded',
  'Authorization' => '••••••'
));
$request->addPostParameter(array(
  'account_type' =>
    '49126a8f-34b5-4928-9ede-c873a1325467',
  'name' => 'Savings',
  'initial_balance' => '15000',
  'colour_code' => ''
));
try {
  $response = $request->send();
  if ($response->getStatus() == 200) {
    echo $response->getBody();
  }
  else {
    echo 'Unexpected HTTP status: ' .
        $response->getStatus() . ' ' .
    $response->getReasonPhrase();
  }
}
catch(HTTP_Request2_Exception $e) {
  echo 'Error: ' . $e->getMessage();
}
```

Database



```sql
use monetdb;
select * from accounts;
```

id	uuid	user_id	account_type_id	currency_id	name	initial_balance	colour_code	active	created_at	updated_at	deleted_at
1	ed005dd9-862a-4a50-9900-cc64851e4213	24	4	105	Savings Account	1500.00	NULL	1	2025-03-12 04:23:45	2025-03-12 04:46:31	2025-03-12 04:46
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	12:52:54	use monetdb	0 row(s) affected	0.000 sec
2	12:52:54	select * from accounts LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Query Completed

CHALLENGES AND SOLUTIONS

1. Issue: Incorrect API Endpoints

Challenge:

- You may enter the wrong API URL, leading to errors like 404 Not Found.
- Example: Sending requests to /user instead of /api/user.

Solution:

- Double-check the API documentation.
- Ensure the correct base URL and endpoints are used.

---

2. Issue: Incorrect HTTP Methods

Challenge:

- Using the wrong method (e.g., GET instead of POST).
- Sending a PUT request for creating instead of updating.

Solution:

- Follow CRUD principles:
  - Create → POST
  - Read → GET
  - Update → PUT / PATCH
  - Delete → DELETE

---

3. Issue: Missing or Incorrect Headers

Challenge:

- API requires headers like Content-Type: application/json, but they are missing.
- Authorization headers are required but not included.

Solution:

- Add required headers in Postman under the Headers tab.
- For authentication, include tokens using Authorization settings.

---

4. Issue: Incorrect JSON Data Format

Challenge:

- Sending malformed JSON causes 400 Bad Request errors.
- Example of incorrect JSON:
- { "name": "John, "age": 25 }

Solution:

- Always validate JSON before sending.
- Use Postman's Prettify feature to format JSON properly.

5. Issue: Database Not Updating

Challenge:

- POST or PUT requests succeed but do not update the database.
- API might return a success response, but changes don’t reflect.

Solution:

- Check server logs for errors.
- Ensure API is correctly handling data and committing changes to the database.

6. Issue: CORS Policy Errors

Challenge:

- If testing with a frontend, you might get CORS (Cross-Origin Resource Sharing) issues.

Solution:

- Enable CORS on the backend.
- Use a CORS extension in the browser for local testing.

7. Issue: Authentication & Authorization Errors

Challenge:

- API requires authentication, but token is missing or expired.

Solution:

- Obtain a fresh authentication token.

- Use Postman’s Authorization tab to add API keys, JWT tokens, or OAuth credentials.

8. Issue: Unexpected API Response

Challenge:

- The API returns an unexpected response (wrong data or errors).

Solution:

- Debug using Postman’s Console (Ctrl + Alt + C) to inspect request details.
- Compare request and response data.