

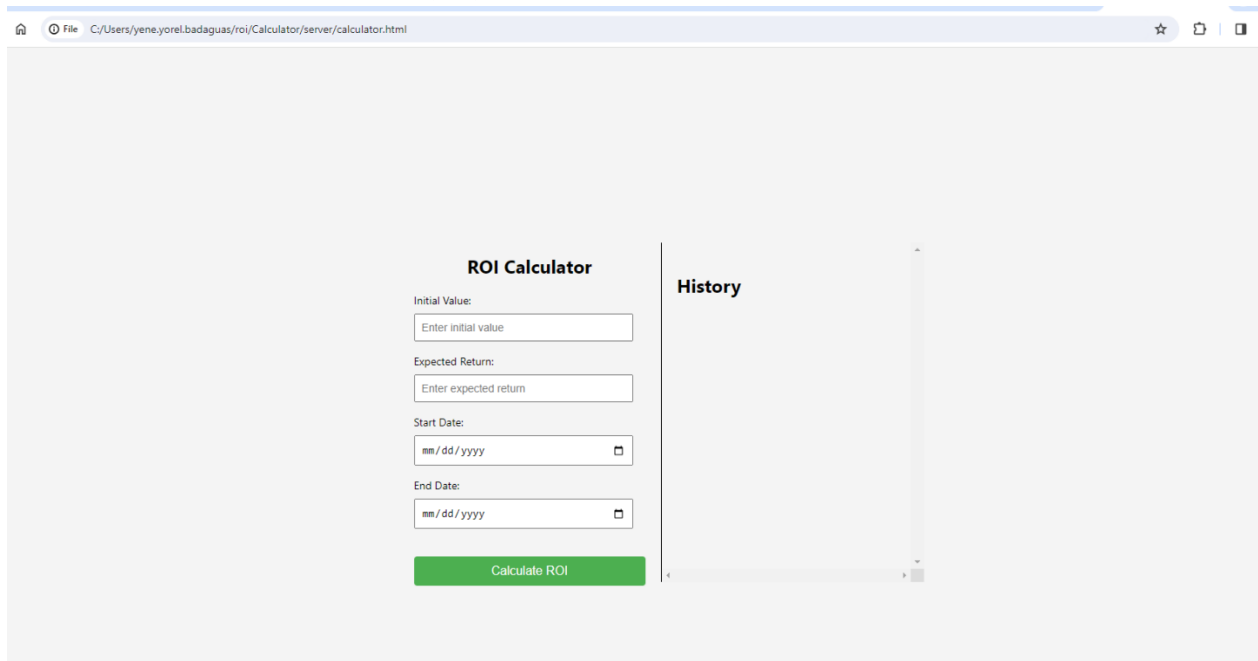
DOCUMENTATION

ROI Calculator – Technical Assessment

1. ROI Calculator Form (.html)

The 'Calculator.html' file serves as the main entry point for our web application. It also includes Javascript Functions:

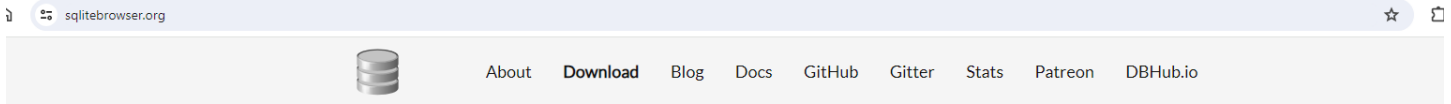
- previousCalculations()
 - Description: This function handles the display of previous calculations on the web page.
 - Usage: Call this function to populate and display previous calculation results.
- calculateROI()
 - Description: This function calculates the Return on Investment (ROI) based on user input.
 - Usage: Trigger this function when new input is provided, and it will update the ROI results on the page.



The screenshot shows a web browser window with the address bar displaying 'C:/Users/yene.yorel.badaguas/roi/Calculator/server/calculator.html'. The main content area is divided into two sections. On the left, titled 'ROI Calculator', there are four input fields: 'Initial Value:' with a placeholder 'Enter initial value', 'Expected Return:' with a placeholder 'Enter expected return', 'Start Date:' with a placeholder 'mm / dd / yyyy' and a calendar icon, and 'End Date:' with a placeholder 'mm / dd / yyyy' and a calendar icon. Below these fields is a green button labeled 'Calculate ROI'. On the right, titled 'History', there is a vertical list area that is currently empty, with a scrollbar on the right side.

2. DB Browser for SQLite

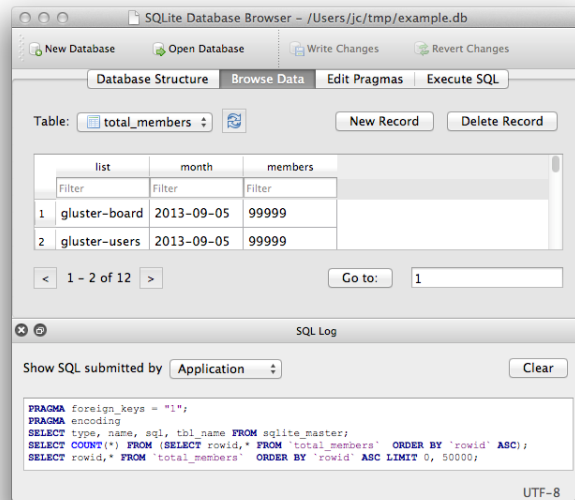
- Install DB Browser for SQLite on your machine (<https://sqlitebrowser.org/dl/>).



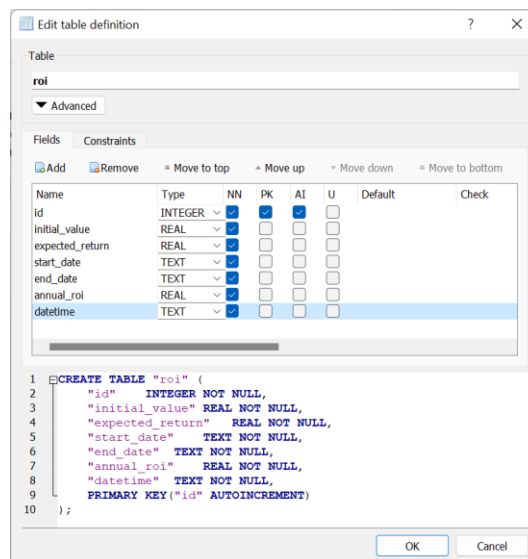
DB Browser for SQLite

The Official home of the DB Browser for SQLite

Screenshot



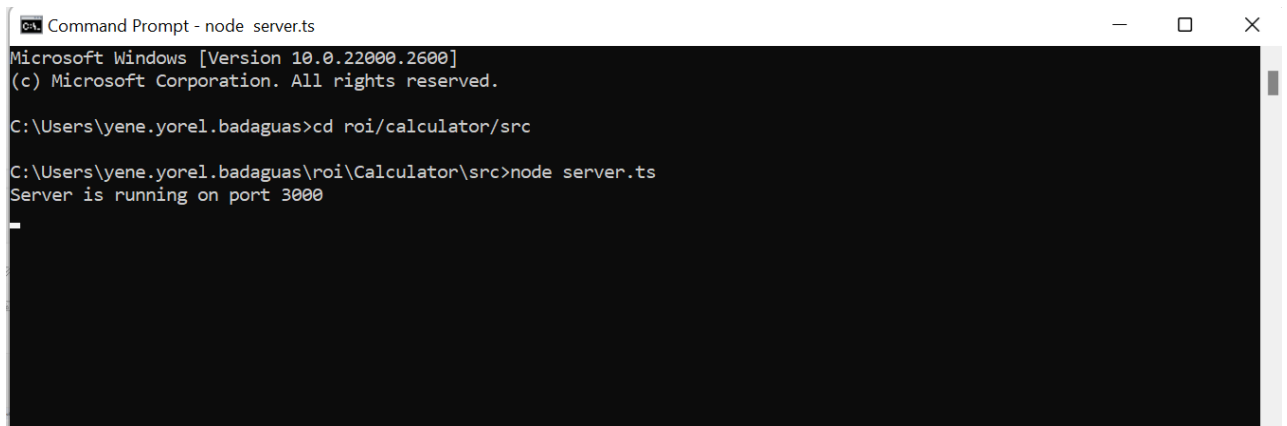
- Create new Database
 - Click on “New Database” to create a new SQLite database file.
 - Save the database file with a desired name, such as ‘roi_data.db’
- Create ‘roi’ Table
 - In the “Database Structure” tab, click on “Create a new table...”
 - Set the table name as ‘roi’
 - Define the columns for ‘roi’ table:



- Click on “Ok” to create the ‘roi’ table.

3. Setup and Run the App Locally

- Installation
 - Make sure you have Node.js and npm install. Then, install the 'sqlite3' package: 'npm install sqlite3'
 - Navigate to the project directory: 'cd roi/calculator/src'
 - Install dependencies: npm install
- run this command to your CMD "node server.ts". "Server is running on port 3000" message will display on your CMD

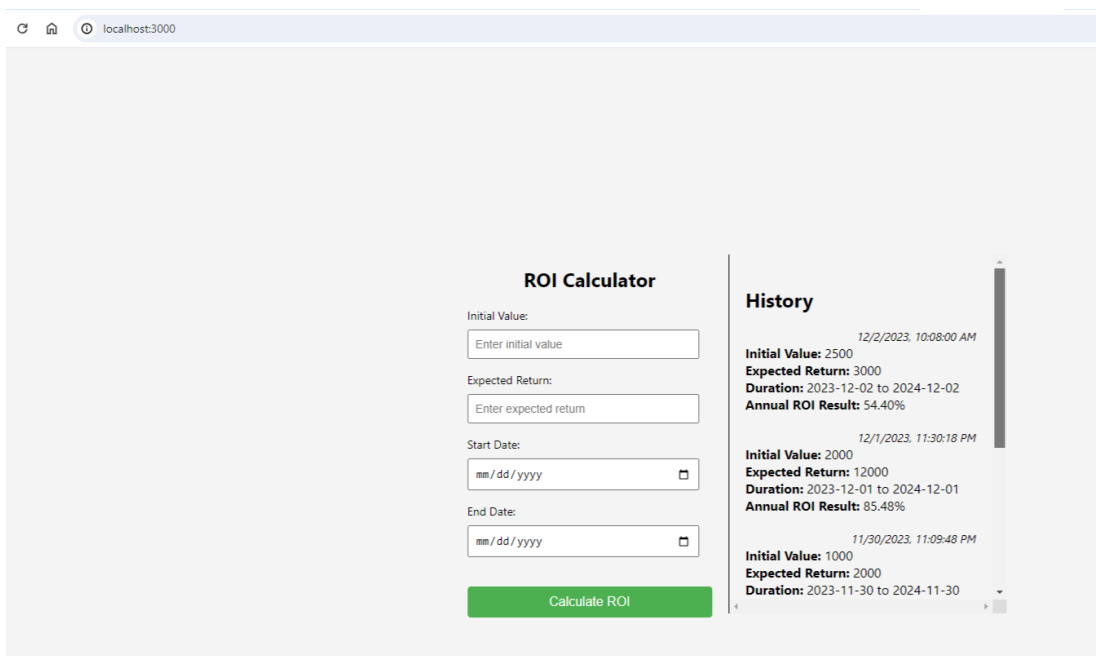


```
Command Prompt - node server.ts
Microsoft Windows [Version 10.0.22000.2600]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yene.yorel.badaguas>cd roi/calculator/src

C:\Users\yene.yorel.badaguas\roi\Calculator\src>node server.ts
Server is running on port 3000
```

- Open the application in your browser (<http://localhost:3000>)



ROI Calculator

Initial Value:

Expected Return:

Start Date:

End Date:

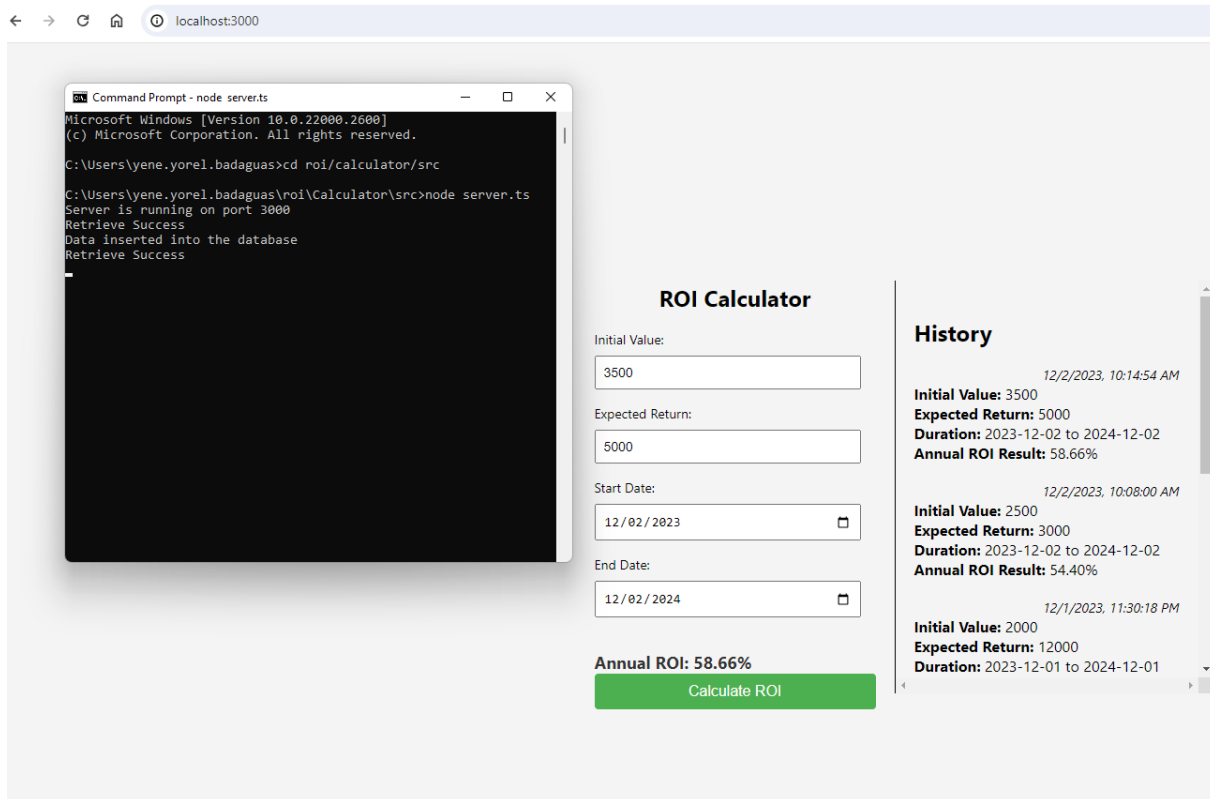
History

12/2/2023, 10:08:00 AM
Initial Value: 2500
Expected Return: 3000
Duration: 2023-12-02 to 2024-12-02
Annual ROI Result: 54.40%

12/1/2023, 11:30:18 PM
Initial Value: 2000
Expected Return: 12000
Duration: 2023-12-01 to 2024-12-01
Annual ROI Result: 85.48%

11/30/2023, 11:09:48 PM
Initial Value: 1000
Expected Return: 2000
Duration: 2023-11-30 to 2024-11-30

- Test the App



4. Supabase

- Supabase Account created (<https://supabase.io/>)
- Create a Supabase Project
 - Log in to your Supabase account.
 - Click on "Create a new project" and follow the prompts.
 - Once the project is created, note the project URL and API key.
- Create a new Table under 'public'.
 - Go to Database tab and select 'public' and click the "+ New table" button.

Create a new table under **public**

Name

Description

☒ **Enable Row Level Security (RLS)** Recommended
 Restrict access to your table by enabling RLS and writing Postgres policies.

① **Policies are required to query data**
 You need to write an access policy before you can query data from this table. Without a policy, querying this table will result in an empty array of results.
 You can create policies after you create this table.

[RLS Documentation](#)

☐ **Enable Realtime**
 Broadcast changes on this table to authorized subscribers

Columns Import data via spreadsheet

Name	Type	Default Value	Primary
<input type="text" value="id"/>	# int8	NULL	<input checked="" type="checkbox"/>
<input type="text" value="initial_value"/>	# float8	NULL	<input type="checkbox"/>
<input type="text" value="expected_reti"/>	# float8	NULL	<input type="checkbox"/>
<input type="text" value="start_date"/>	📅 date	NULL	<input type="checkbox"/>
<input type="text" value="end_date"/>	📅 date	NULL	<input type="checkbox"/>
<input type="text" value="annual_roi"/>	# float8	NULL	<input type="checkbox"/>
<input type="text" value="datetime"/>	📅 timestamp	NULL	<input type="checkbox"/>

[Add column](#) [Learn more about data types](#)

5. Migration to Supabase

- Go to the Migration folder and open the 'migration.ts' file and update 'supabaseURL' and 'supabaseKey' based on your supabase project.
- Run the Migration Script
 - Change the directory to your project directory. Check screen shot below

```
Command Prompt
Microsoft Windows [Version 10.0.22000.2600]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yene.yorel.badaguas>cd roi/calculator/migration
C:\Users\yene.yorel.badaguas\roi\Calculator\migration>
```

- Execute the migration script using: 'ts-node migration.ts'. All records from sqlite were displayed and migrated to supabase.

```
Select Command Prompt
C:\Users\yene.yorel.badaguas\roi\Calculator\migration>ts-node migration.ts
etched data from SQLite: [
  {
    id: 1,
    initial_value: 1000,
    expected_return: 2000,
    start_date: "2023-11-30",
    end_date: "2024-11-30",
    annual_roi: 66.4845173041894,
    datetime: "11/30/2023, 6:00:29 PM"
  },
  {
    id: 2,
    initial_value: 1000,
    expected_return: 2000,
    start_date: "2023-11-30",
    end_date: "2024-11-30",
    annual_roi: 66.4845173041894,
    datetime: "11/30/2023, 11:00:48 PM"
  },
  {
    id: 3,
    initial_value: 1000,
    expected_return: 2000,
    start_date: "2023-11-30",
    end_date: "2024-11-30",
    annual_roi: 66.4845173041894,
    datetime: "11/30/2023, 11:00:48 PM"
  },
  {
    id: 4,
    initial_value: 2000,
    expected_return: 12000,
    start_date: "2023-12-01",
    end_date: "2024-12-01",
    annual_roi: 85.480093676815,
    datetime: "12/1/2023, 11:00:18 PM"
  }
]
migration to Supabase successful: null
C:\Users\yene.yorel.badaguas\roi\Calculator\migration>
```

- Verify Migration
 - Check the Supabase dashboard to verify that the data has been successfully migrated to the target table.
 - Run queries on the Supabase SQL editor to inspect the migrated data.

supabase.com/dashboard/project/wmbmmsasrgnyvhdzjx/sql/87f62bd4-8f4f-47bc-b572-2ec879e872b6

yeneyorel's Org (Free) / ROI Calculator

SQL Editor

+ New query

Templates

Quickstarts

Your queries

Select all from public.roi

Select all users

Ask Supabase AI to modify your query

```
1 select * from public.roi
```

Results (4)

id	initial_value	expected_return	start_date	end_date	annual_roi	datetime
1	1000	2000	"2023-11-30"	"2024-11-30"	66.4845173041894	"2023-11-30 18:00:29"
2	1000	2000	"2023-11-30"	"2024-11-30"	66.4845173041894	"2023-11-30 23:00:48"
3	1000	2000	"2023-11-30"	"2024-11-30"	66.4845173041894	"2023-11-30 23:00:48"
4	2000	12000	"2023-12-01"	"2024-12-01"	85.480093676815	"2023-12-01 23:00:18"