

A.1 - Project Report: Sections 1 and 2

HSC Software Engineering AT3 Project Report

Project Title: Advanced Python-powered calculator PWA

Student Name: Ethan Tan

Student Number: 39025345

Class: Year 12 Software Engineering

Submission Date: 10/06/2025

GitHub Repository:

<https://github.com/L-cpu-cpu/HSC-SE-Final-Project---Advanced-Python-Calculator>

Section 1: Justification of Development Approach and Planning

1.1 Problem Definition

Often, when without a physical calculator, students resort to the Google-based calculator on their laptops. However, the calculator's functionality is limited and not suitable for calculating formulae and using results from multiple calculations. This project aims to create a PWA (Progressive Web App) calculator that will address these issues. Having such a PWA will allow high school students to complete math work intuitively without a scientific calculator. A bigger screen and colour will allow for better visibility and readability of results.

1.2 Development Methodology

As a result of clear functional requirements and low chances of extra functionality beyond those stated, the waterfall approach to software development will be used. The project's components will be approached in a bottom-top manner, creating individual components before integrating them all together to result in the final product. In order to refine the project and remove errors, an approach similar to agile development will be used for the refinement stage of the project.

1.3 Planning Tools

Planning tools used to coordinate and record the development process are Gantt charts, process diaries to record project progress and a GitHub repository for version control of the project.

Gantt chart: A visual representation created on Google Docs in a timeline fashion, outlining the main stages of the project. The chart is updated in response to unexpected changes.

Task Name	Dec 2024	Jan 2025	Feb	Mar	Apr	May	Jun wk1	Jun wk2
Project Proposal	█							
Requirements definition			█		█	█		
Design				█			█	
Documentation & Report	█		█	█			█	
Dependencies and environment						█		
Database (SQLite)						█		
Backend (Python)						█		
Frontend (HTML and CSS)						█	█	
Backend (JavaScript)						█	█	
Testing							█	█
Submission								█
Presentation								█

Process Diary: Entries showing tasks, the stage of the development process, date and status

Activity	Stage	Date	Tool	Status
Project proposal	Requirements definition	25/10/2024	Google Docs	Completed
Project report created	Documentation	5/2/2025	Google Docs	Completed
Requirements & Specifications	Requirements definition and Determining Specifications	5/2/2025	Google Docs	Completed
Structure chart created	Design	5/2/2025	Draw.io	Completed

Level 0 DFD created	Design	21/4/2025	Draw.io	Completed
Begun working on Level 1 DFD	Design	21/4/2025	Draw.io	Completed (25/4/2025)
Begun working on Level 2 DFD	Design	26/4/2025	Draw.io	Completed (26/4/2025)
Decision Trees started	Design	28/4/2025	Draw.io	Completed
Documentation format updated	Other	7/5/2025	Google Docs	Completed
Additional functionality added to include software automation, PWA and secure software architecture	Requirements definition	7/5/2025	Google Docs	Completed
Virtual environment created	Implementation	16/5/2025	VsCode	Completed
Formula list module file	Implementation	16/5/2025	VsCode	Completed
Database manipulation module	Implementation	26/5/2025	VsCode	Revised (1/6/2025)
Calculation Module	Implementation	28/5/2025	VsCode	Revised (1/6/2025)
User input formatting module	Implementation	28/5/2025	VsCode	Revised (1/6/2025)
Bare bones UI made	Implementation	31/5/2025	VsCode	Completed (1/6/2025)
Integration of modules with HTML and Python Flask	Implementation	31/5/2025	VsCode	Completed (3/6/2025)
Addition of QOL for UI to improve UX	Implementation	2/6/2025	VsCode	Completed (3/6/2025)
Addition of	Implementation	4/6/2025	VsCode	Completed

sorting and searching for saved results, as well as a button to remove saved results				
Fixed inputs so they do not accept unsafe characters (<, >, “, ’)	Implementation	4/6/2025	VsCode	Completed
Implemented matplotlib to display a graph and a plot of the regression equation.	Implementation	6/6/2025	VsCode	Completed
Styled HTML files, inputs, dropdowns and buttons	Implementation	6/6/2025	VsCode	Completed (7/6/2025)
Added more formulae to the Physics category	Implementation	7/6/2025	VsCode	Completed
ReadMe updated	Documentation	7/6/2025	GitHub	Completed
Updated design diagrams	Design	7/7/2025	Draw.io	Completed (8/6/2025)
Testing and evaluation	Testing and evaluation	9/6/2025	Google Chrome	Completed

1.4 Functional Requirements

Function:

- Perform a series of basic mathematical functions (+, -, *, /) in a single user input
- Temporarily store calculation results for use in future calculations, similar to the ‘ans’ button on physical calculators
- Allow the user to perform advanced mathematical functions, including but not limited to:
 - Trigonometry
 - Roots

- Logarithms
 - Exponentials
 - Factorials
- Allow the user to input values for a variety of formulas as a separate mode of operation, including but not limited to:
 - Areas, perimeters, volumes and surface areas of simple geometric shapes
 - Physics formulae such as Newton's second law or Einstein's $E = mc^2$
 - Trigonometry and geometric formulae, such as SOH CAH TOA
- Allow for the storing of calculation results along with their input in a database that can be retrieved for calculations
- Allow the user to display calculation results to an input number of significant digits if needed
- Allow for the input of integers, decimals and negative numbers
- Allow the user to temporarily store calculation results into variables from a-z that are removed at the end of the session
- Allow for a user to input a table of values to find a regression equation

Output:

- Output the result of calculations done by the user to a set number of significant digits
- Display the different modes of the calculator in a simple and easy-to-navigate menu
- Display a user's stored calculations and allow for the viewing of the calculation and result

Usage:

- Allow the user to choose between calculator modes in a menu, allowing for the selection of options and backtracking
- The user should be able to input numbers, operations and functions within a single input
- In a separate mode of operation, the user can input variables for a chosen formula
- A user can save the current calculation result in a database entry along with a user input keyword for identification when retrieving in calculation mode
- A user can use the result of the previous calculation in the current input through the keyword 'ans'
- A user can temporarily store results in a variable for later use in a session

1.5 Success Criteria (Basic Overview)

Criterion	Met?
The user can input a calculation, and it will be properly calculated (i.e the right result is displayed)	Yes
The user can save the result of a calculation as a session variable	Yes

The user can save the result of a calculation along with a keyword into a database entry	Yes
The user can click a menu option and properly view all saved results	Yes
The user can input a keyword when calculating, and the corresponding saved result/variable will be substituted in	Yes
The user can select a formula, input the values, and the correct result will be displayed	Yes
The user can change the number of significant digits of the displayed results	Yes
The user can successfully input a table of values and receive a graph and a regression equation	Yes
Advanced math functions, such as trigonometry and roots, can be input properly and work	Yes
Input forms are resistant to XSS and SQL injection	Yes

Section 2: Justification of Tools and Resources Used

2.1 Frontend Tools

- **HTML/CSS/t:** Core languages for UI.
 - HTML renders webpages and is widely used on most browsers; compatibility with Jinja2 allows for easy styling to be applied to all pages.
 - CSS provides class-based styling with a wide variety of options, allowing for good aesthetic value. The inclusion of animations also opens up more opportunities for styling.
- **Service Worker and Manifest:** PWA implementation
 - Service workers and manifests assist with caching the files of the PWA, allowing it to be rendered offline

2.2 Backend Tools

- **Python:** For main functionality

- A large library of modules allows for great flexibility. For example, the math module can be used to allow for mathematical functions and constants
 - The ability to import the functionality of Python files into another allows for easy modularisation of the system
 - The Flask module allows for the dynamic updating of content on pages, as well as passing values between different pages
- **SQLite:** For database functionality
 - Allows for the creation and manipulation of databases without the need for an external database, like with MAMP
 - Works with Python, with the sqlite3 module being the only thing needed to import
 - Allows Python to use SQL requests to interact with the database
- **JavaScript:**
 - Allows for the validation of inputs
 - Allows for live updating of pages by being able to send or retrieve data in the background
 - Can work with HTML to render dynamic content

2.3 Version Control and Workflow

- **GitHub:** For version control
 - Github gives ease in version control with the ability to fork out to apply changes, the ability to go to past versions, and view the changes made to files
- **Google Docs:** For process diary, Gantt chart and overall documentation
 - Google Docs provides a user-friendly process for documenting in digital formats.
 - Google Docs can be exported as PDF files, allowing for documentation to be included in the final submission
- **Draw.io:** For all diagrams
 - A simple UI and basic tools allow for easily drafting diagrams, with edits being automatically applied to the file.

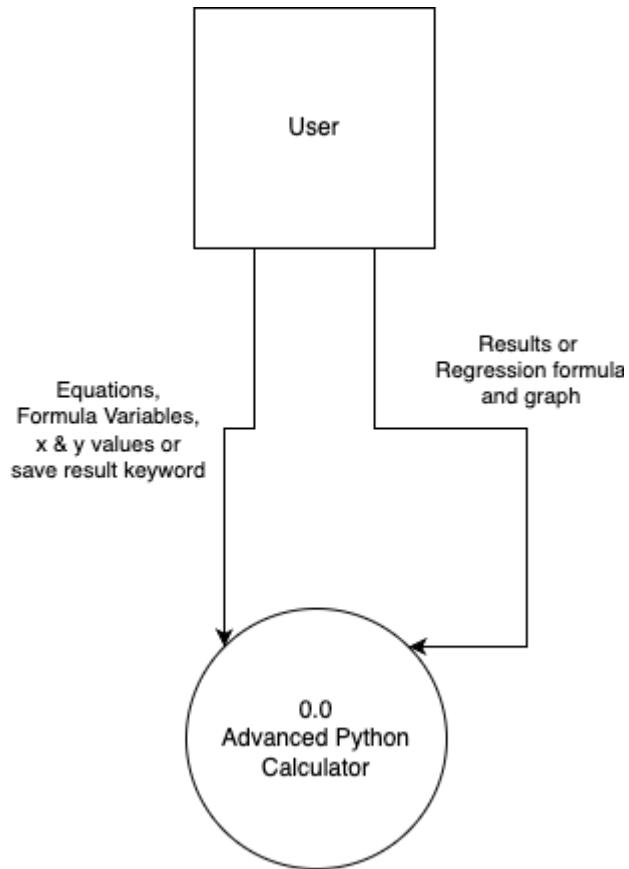
A.2: Diagrams

System Modelling Diagrams

- Context diagram / Level 0 DFD
- Level 1 & 2 DFD
- Structure chart
- Data dictionary
- Wireframe storyboard

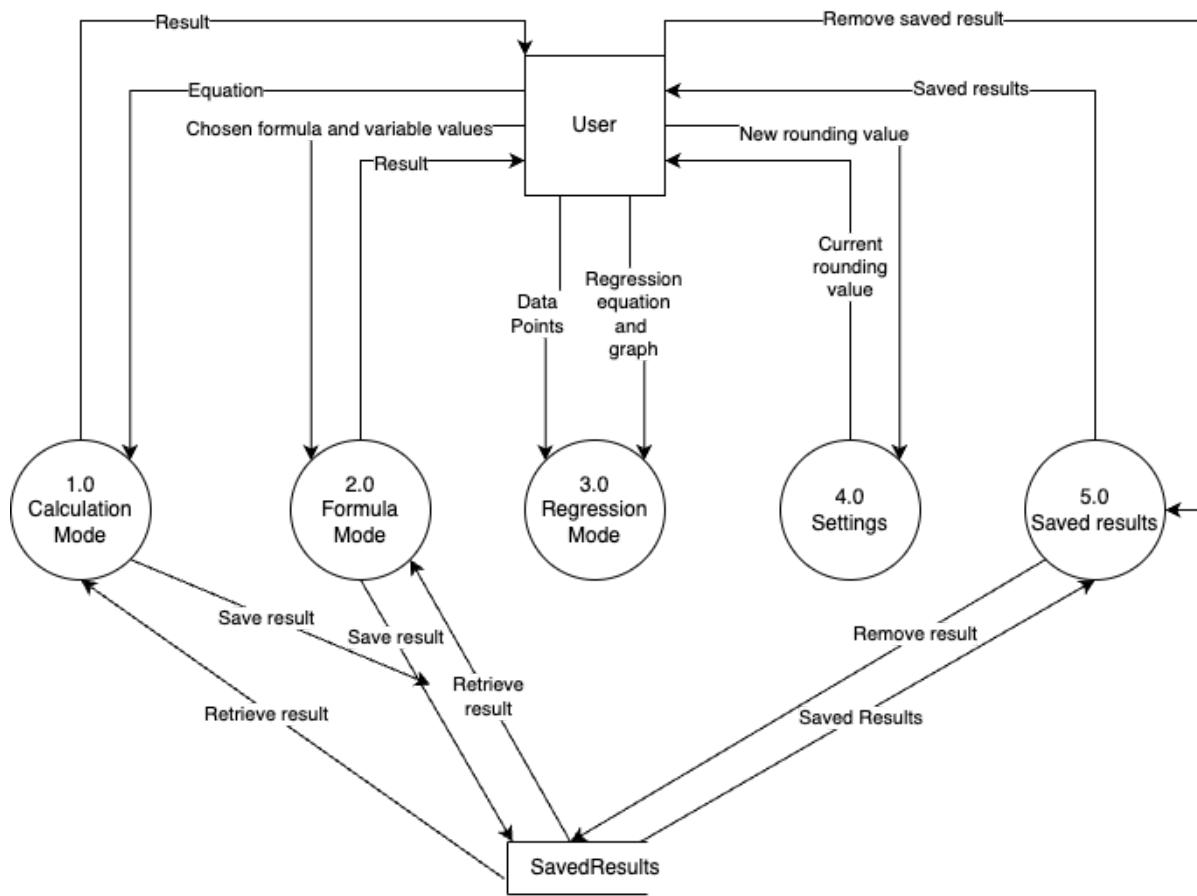
Diagrams

Context diagram (Level 0 DFD)



Showing overall inputs and outputs

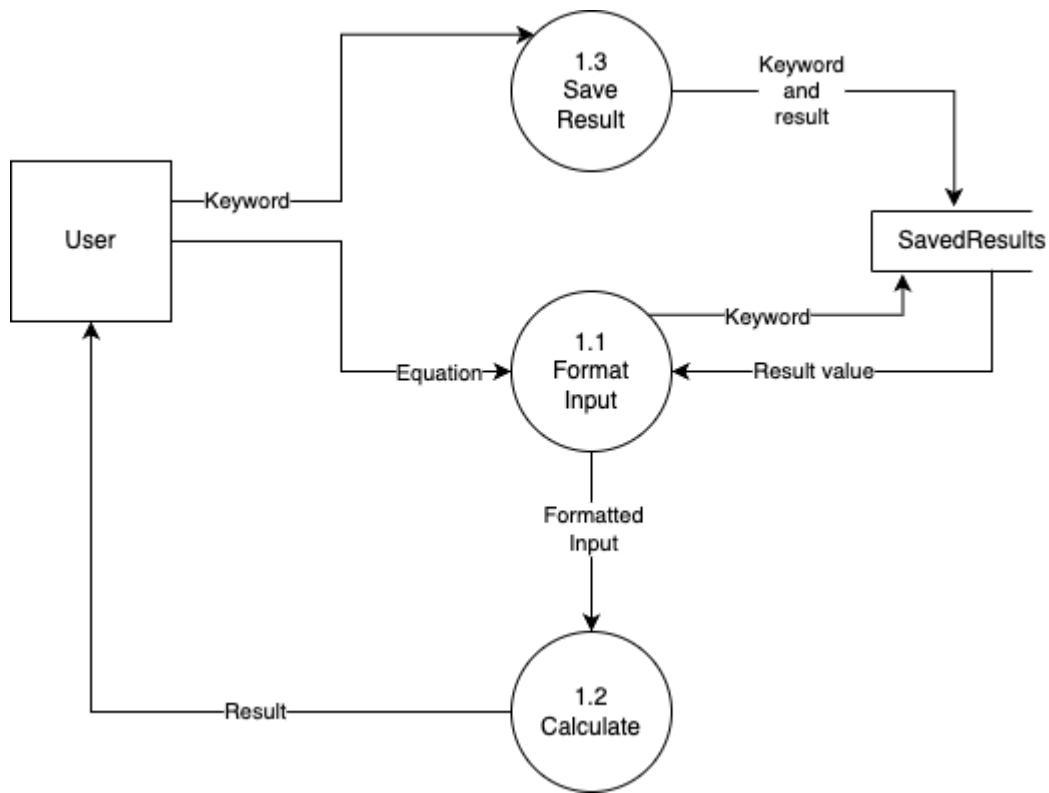
Level 1 DFD



Showing inputs and outputs for each mode, as well as showing the interactions between functions and SavedResults

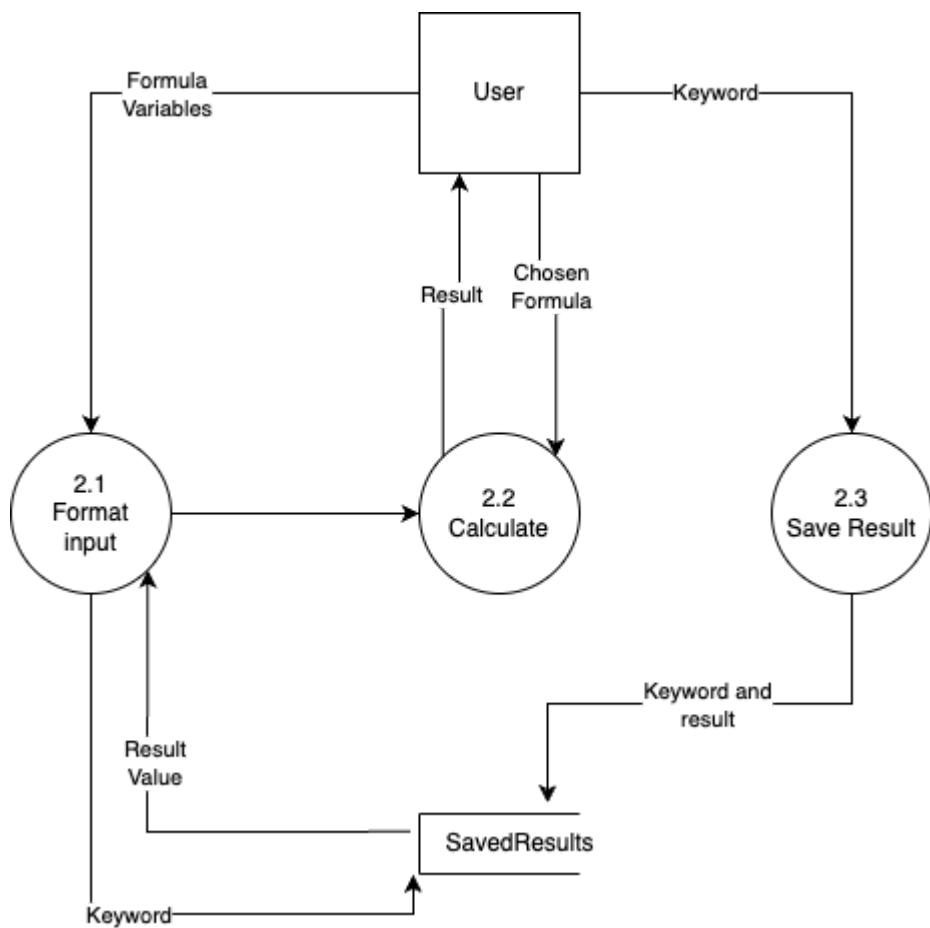
Level 2 DFD

1.0 Calculation mode



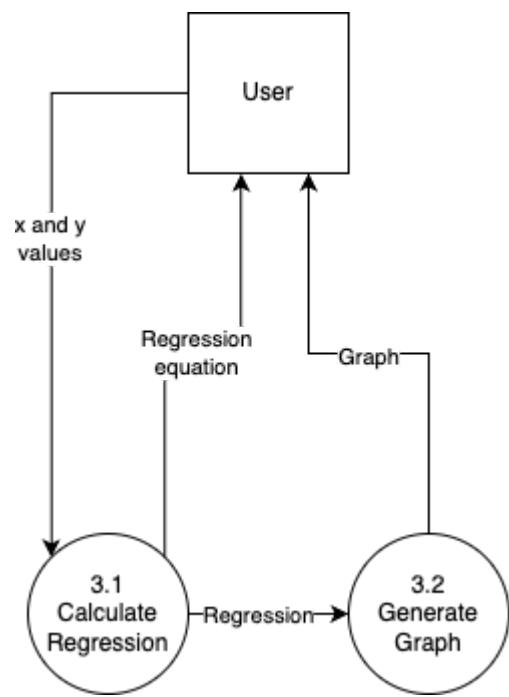
Showing the overall function of the calculation mode and its interaction with SavedResults

2.0 Formula mode



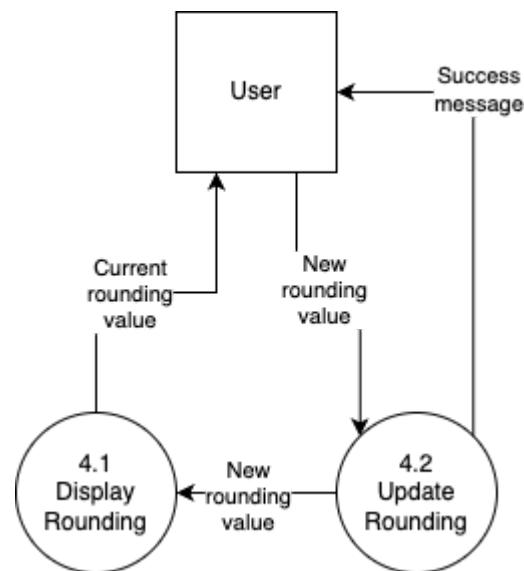
Showing the overall function of the formula mode and its interaction with SavedResults

3.0 Regression mode



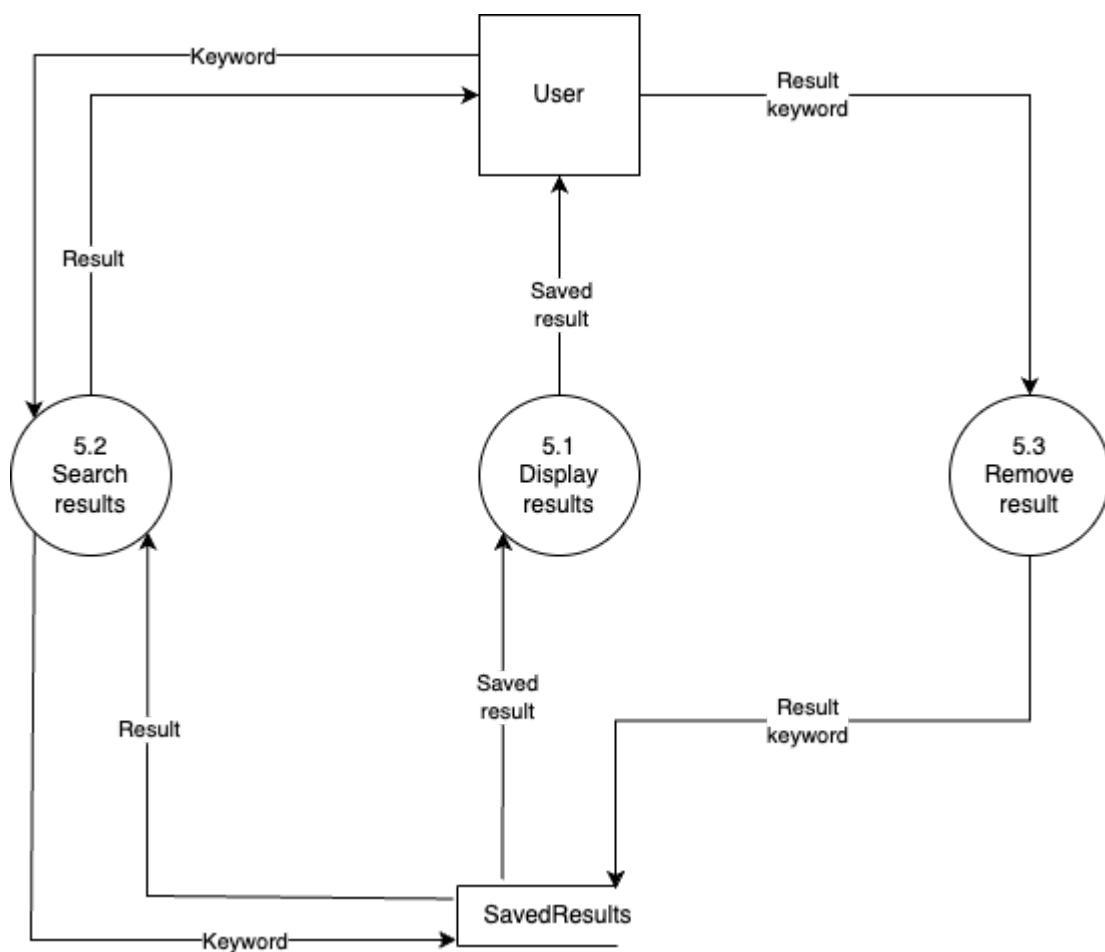
Showing the overall function of the regression mode.

4.0 Settings



Showing the overall function of the settings

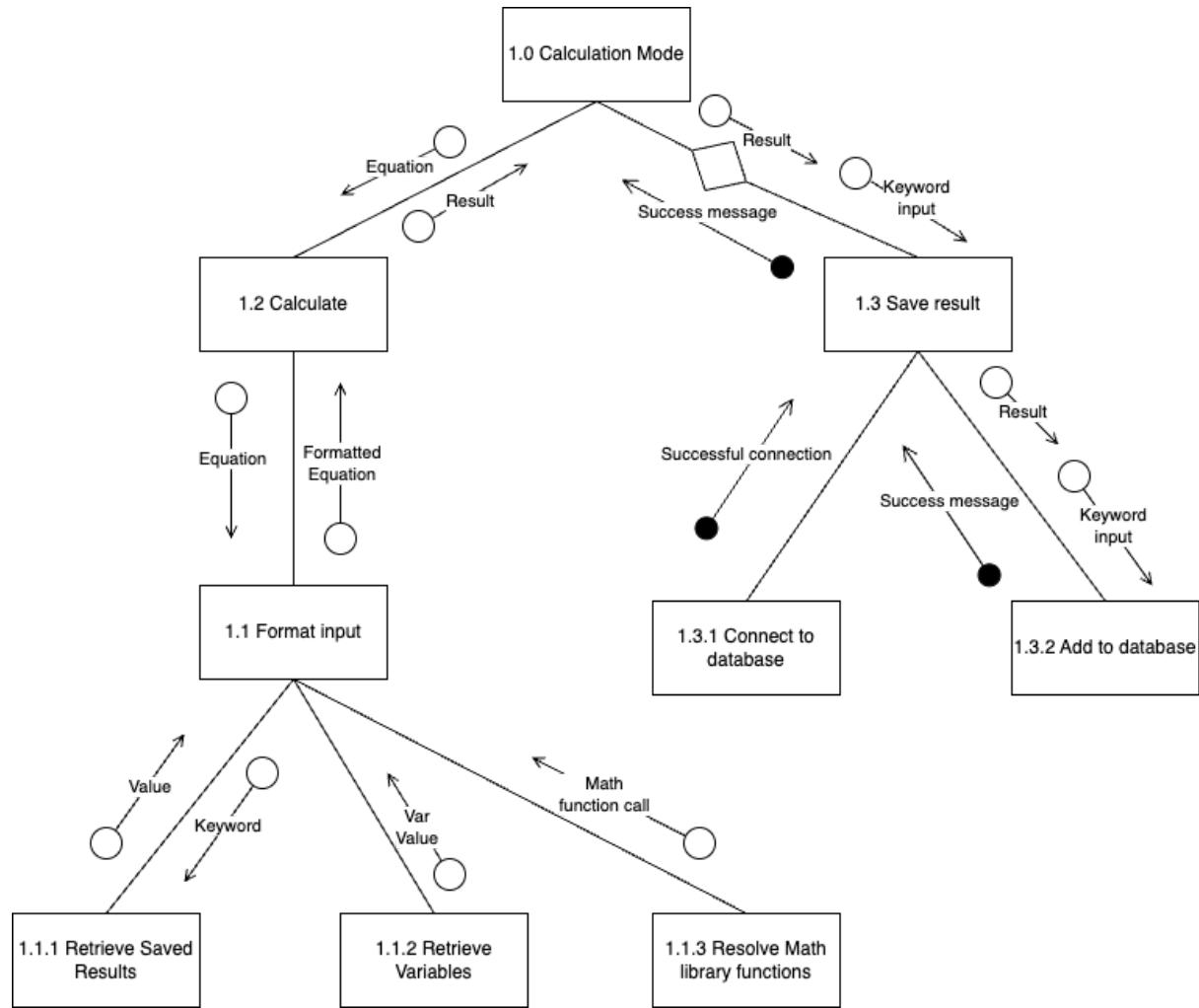
5.0 View saved results



Showing the overall function of viewing saved results

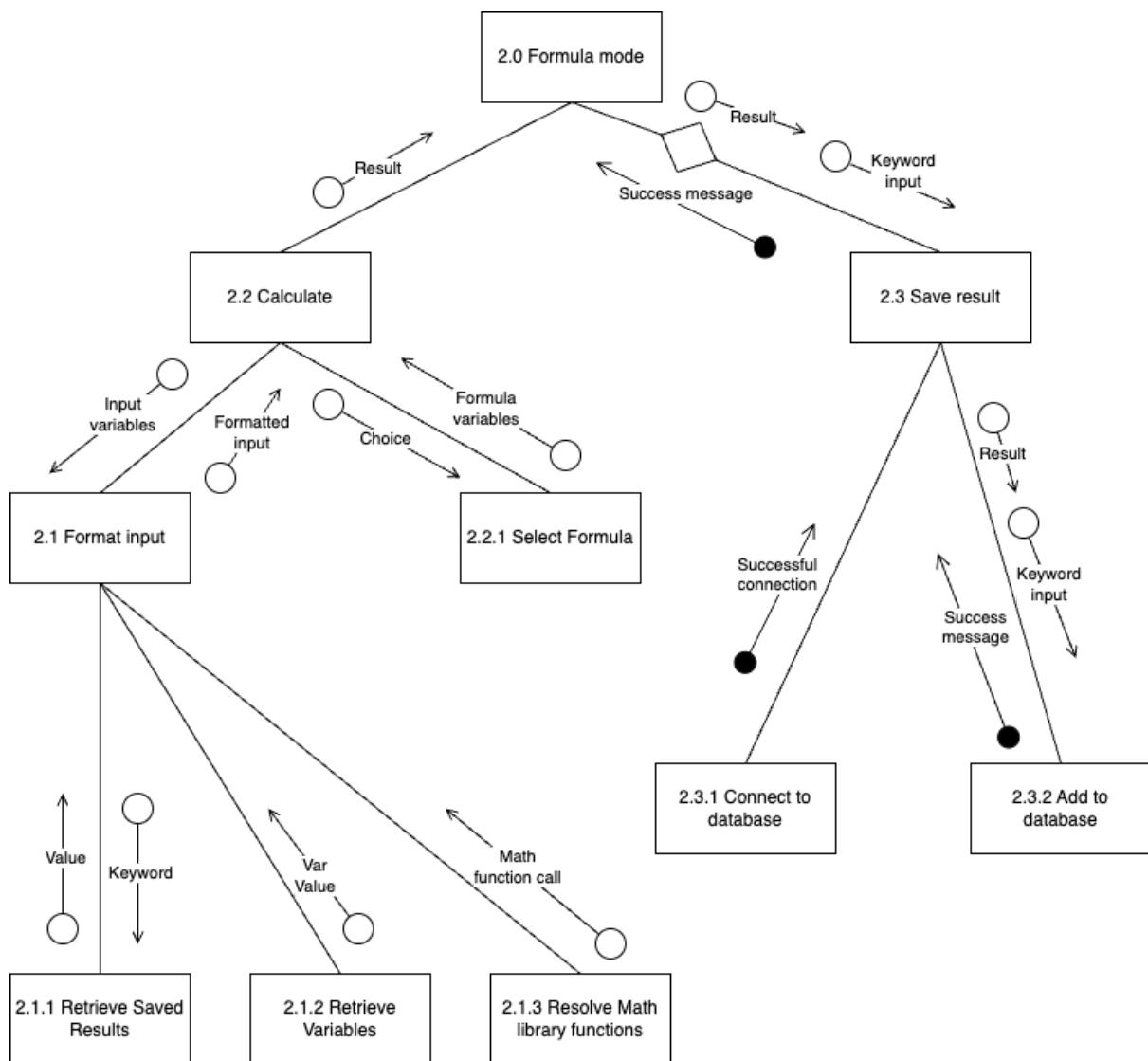
Structure charts

1.0 Calculation mode



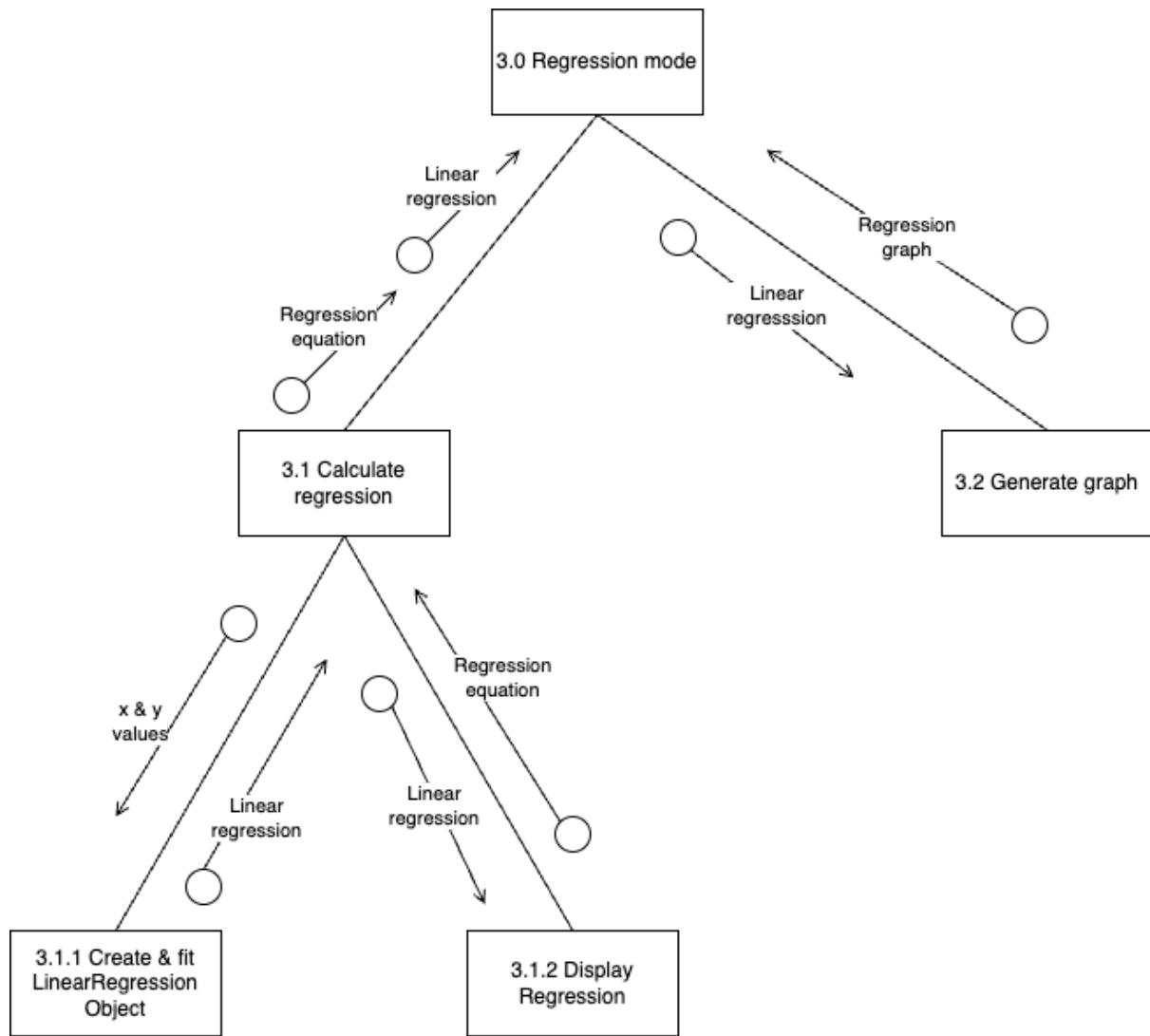
A simplified overview of the Calculation mode

2.0 Formula mode



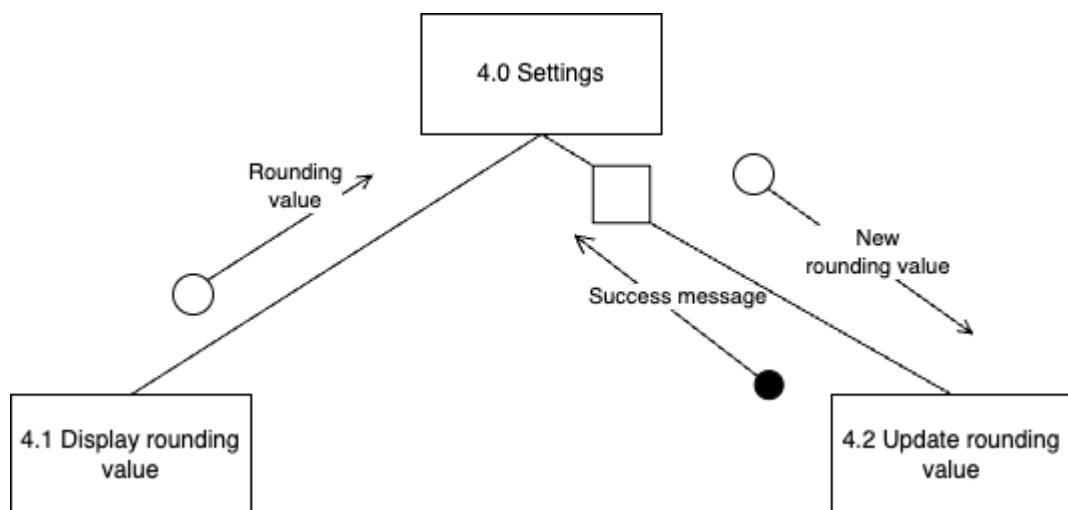
A simplified overview of the Formula mode

3.0 Regression mode



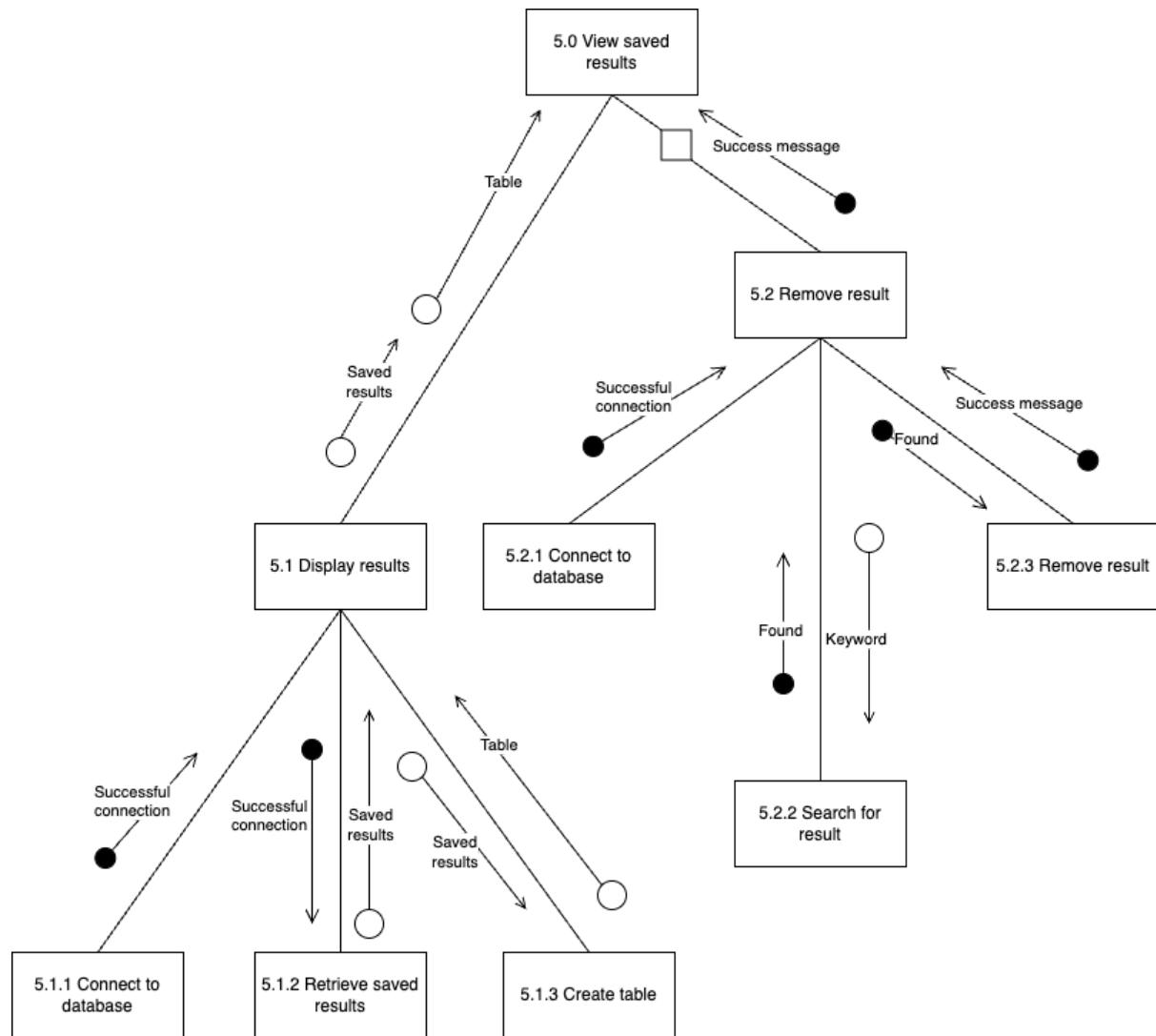
An overview of the regression mode

4.0 Settings



An overview of how a user can view and change the value, which results will round to

5.0 View saved results



An overview of how saved results are displayed and how a user would remove results

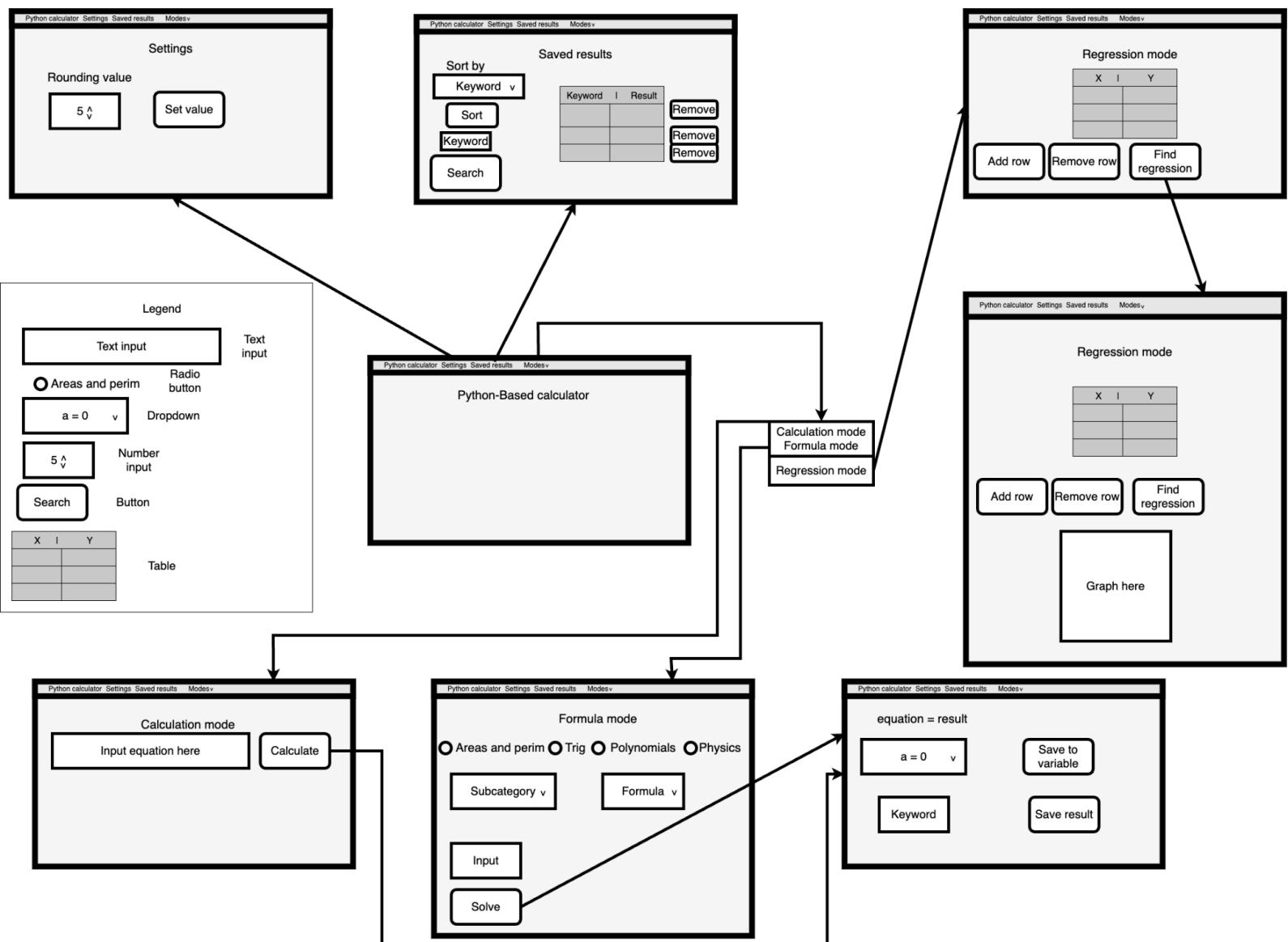
Data Dictionary:

savedResults.db

Table: savedResults

Field	Data Type	Description	Example	Constraints
keyword	TEXT	User-input label for result	"MyResult"	Primary Key, Not Null
result	REAL	Value of result	20.0	Not Null

Wireframe storyboard



A.3 Project Report: Evaluation

No.	Criterion	Status	Explanation
1	Calculation mode functionality	Achieved	The user can input a calculation, and it will be properly calculated (i.e the right result is displayed)
The equation used for testing is $1 + 1 \times (2 + 5)^2$ The expected result is 50			

1.

Python calculator Settings Saved Results Modes ▾

Calculation Mode

Input equation here

Submit

How to Use: Input your equation and click calculate to calculate. You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: a + 2 * (SIN(20))^3

2.

Python calculator Settings Saved Results Modes ▾

Calculation Mode

Input equation here

1 + 1 * (2 + 5)^2

Submit

How to Use: Input your equation and click calculate to calculate. You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: a + 2 * (SIN(20))^3

3.

Python calculator Settings Saved Results Modes ▾

$1 + 1 * (2 + 5)^2 =$
50

a = 0 ▾ Save to Variable

Keyword to save result

Save to Table

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

2	Session variable	Achieved	The user can save the result of a calculation as a session variable
---	------------------	----------	---

1.

Python calculator Settings Saved Results Modes ▾

$1 + 1 * (2 + 5)^2 =$
50

a = 0 ▾ Save to Variable

Keyword to save result

Save to Table

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

2.

Python calculator Settings Saved Results Modes ▾

a = 0
b = 0
c = 0
d = 0
e = 0
f = 0
g = 0
h = 0
i = 0
j = 0
k = 0
l = 0
m = 0
n = 0
o = 0
p = 0
q = 0
r = 0
s = 0
t = 0
u = 0
v = 0
w = 0
x = 0
y = 0
z = 0

(2 + 5)² =
50

Save to Variable
Save to Table
Keyword to save result

How to Use: Save the result permanently by entering a keyword and clicking Save to Variable or Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

3.

Python calculator Settings Saved Results Modes ▾

1 + 1 * (2 + 5)² =
50

g = 50 ▾ Save to Variable
Keyword to save result
Save to Table
Saved result to variable!

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

3	Database saving	Achieved	The user can save the result of a calculation along with a keyword into a database entry
---	-----------------	----------	--

1.

Python calculator Settings Saved Results Modes ▾

$1 + 1 * (2 + 5)^2 =$
50

a = 0 ▾ Save to Variable

Keyword to save result

Save to Table

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

2.

Python calculator Settings Saved Results Modes ▾

$1 + 1 * (2 + 5)^2 =$
50

a = 0 ▾ Save to Variable

Keyword to save result

Test8

Save to Table

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

3.

Python calculator Settings Saved Results Modes ▾

$1 + 1 * (2 + 5)^2 =$
50

a = 0 Save to Variable
 Keyword to save result
 Test8
 Save to Table
 Saved result!

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

4.

	keyword	result
	PRIMARY KEY	REAL
1	Test1	20.0
2	Test2	30.0
3	Test3	50.0
4	Test4	72.0
5	Test5	100.0
6	Test8	50.0

```
INSERT INTO `savedResults` DEFAULT VALUES
keyword
  1 NULL
result
  1 NULL
✓ Commit ↗ ?
```

PROBLEMS 49 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
source /Users/thefolderoffolders/Documents/HSCSE-FinalProject/virtualEnvironment/bin/activate
/Users/thefolderoffolders/Documents/HSCSE-FinalProject/virtualEnvironment/bin/python /Users/thefolderoffolders/Documents/HSCSE-FinalProject/main.py
thefolderoffolders@GATEWAY-etta6569 HSCSE-FinalProject % source /Users/thefolderoffolders/Documents/HSCSE-FinalProject
virtualEnvironment thefolderoffolders@GATEWAY-etta6569 HSCSE-FinalProject % /Users/thefolderoffolders/Documents/HSCSE-FinalProject/virtualEnvironment/bin/python /Users/thefolderoffolders/Documents/HSCSE-FinalProject/main.py
* Serving Flask app 'main'
* Debug mode: on
[]
```

zsh
zsh
zsh
zsh
zsh
zsh
Python

6 records

4	Saved results	Achieved	The user can view all saved results in a table format
1.			

Python-based Calculator

2.

Saved Results

Sort by: Keyword ▾ Sort

Search by keyword:

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test3	50.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove
Test8	50.0	Remove

3.

5	Sorting	Achieved	The user can sort the saved results table by keyword or value
---	---------	----------	---

1.

Keyword	Result	Action
Test1	20.0	<button>Remove</button>
Test2	30.0	<button>Remove</button>
Test3	50.0	<button>Remove</button>
Test4	72.0	<button>Remove</button>
Test5	100.0	<button>Remove</button>
Test8	50.0	<button>Remove</button>

2.

Python calculator Settings Saved Results Modes ▾

Saved Results

Sort by: ✓ Result Sort

Search by keyword:

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test3	50.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove
Test8	50.0	Remove

3.

Python calculator Settings Saved Results Modes ▾

Saved Results

Sort by: Result Sort

Search by keyword:

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test3	50.0	Remove
Test8	50.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove

6	Searching	Achieved	The user can search for a specific result by keyword
---	-----------	----------	--

1.

Saved Results

Sort by: Keyword ▾ Sort

Search by keyword:

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test3	50.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove
Test8	50.0	Remove

2.

Saved Results

Sort by: Keyword ▾ Sort

Search by keyword:

Test5

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test3	50.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove
Test8	50.0	Remove

3.

Saved Results

Sort by: Keyword Sort

Search by keyword:

Test5

Search

Keyword	Result	Action
Test5	100.0	Remove

How to Use: Enter a keyword to search for a specific result, or use the dropdown to sort the results. Click remove to remove a result.

7	Removal	Achieved	The user can remove a specific result from the database
---	---------	----------	---

1.

Saved Results

Sort by: Keyword Sort

Search by keyword:

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test3	50.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove
Test8	50.0	Remove

2.

Python calculator Settings Saved Results Modes ▾

Saved Results

Sort by: Keyword Sort

Search by keyword:

Search

Keyword	Result	Action
Test1	20.0	<button>Remove</button>
Test2	30.0	<button>Remove</button>
Test4	72.0	<button>Remove</button>
Test5	100.0	<button>Remove</button>
Test8	50.0	<button>Remove</button>

How to Use: Enter a keyword to search for a specific result, or use the dropdown to sort the results. Click remove to remove a result.

3.

EXPLORER

- HSCSE-FINALPROJECT
 - .vscode
 - DataBases
 - savedResults.db
 - Modules
 - __pycache__
 - calculate.py 5
 - dbFunctions.py
 - formattingInput.py
 - formulae.py
 - regression.py
 - variables.py
 - node_modules
 - static
 - templates
 - partials
 - footer.html
 - loadingScreen.html
 - menu.html
 - answer.html
 - calculateFormula.h...
 - displayResult.html
 - formCalcMode.html
 - index.html
 - layout.html
 - regression.html
 - settings.html
 - virtualEnvironment
- main.py 9+
- package-lock.json
- package.json
- security_log.log

SELECT * FROM savedResults		Schema	Query Editor	Auto Reload	Find	Other Tools...
keyword	TEXT	PRIMARY KEY	result	REAL		
1	Test1	20.0				
2	Test2	30.0				
3	Test4	72.0				
4	Test5	100.0				
5	Test8	50.0				

```
INSERT INTO `savedResults` DEFAULT VALUES
keyword
1 NULL
result
1 NULL
Commit
```

```
AS TEXT NUMERIC BLOB NULL DEFAULT
```

```
AS TEXT NUMERIC BLOB NULL DEFAULT
```

```
History ...
```

```
return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args) # type: ignore[no-any-return]
File "/Users/thefolderoffolders/Documents/HSCSE-Finalproject/main.py", line 110, in calculation
    session['result']= calc.calculate(equationInput, session['ans'])
File "/Users/thefolderoffolders/Documents/HSCSE-Finalproject/Modules/calculate.py", line 28, in calculate
    ans = eval(str(equation))
File "<string>", line 1, in <module>
NameError: name 'ans' is not defined

```

```
File "/Users/thefolderoffolders/Documents/HSCSE-Finalproject/virtualEnvironment/bin/python /Users/thefolderoffolders/Documents/HSCSE-Finalproject/main.py
  File "run.py", line 1, in <module>
ModuleNotFoundError: No module named 'run'
```

8	Saved result and variable substitution	Achieved	The user can input a keyword when calculating, and the corresponding saved result/variable will be substituted in (Formula and Calculation mode)
---	--	----------	--

The values of Test1 and Test2 are 20 & 30. Variable b has been set to 50. The answer

should be 100

1.

Python calculator Settings Saved Results Modes ▾

Calculation Mode

Input equation here

Test1 + Test2 + b

How to Use: Input your equation and click calculate to calculate. You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: a + 2 * (SIN(20))^3

2.

Python calculator Settings Saved Results Modes ▾

Test1 + Test2 + b =

100.0

Keyword to save result

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

9	Formula mode functionality	Achieved	The user can select a formula, input the values, and the correct result will be displayed
---	----------------------------	----------	---

The test formula is for a right-angle triangle. Test values for base and height are 20 & 10,

meaning that the result should be 100 ($0.5 * 20 * 10$)

1.

Python calculator Settings Saved Results Modes ▾

Formula Mode

Please select a category to view the available formulas.

Areas and Perimeters Trigonometry Polynomials Physics

Select a formula subcategory: Triangles Select a formula: Right angle Triangles

20
 10

How to Use: Choose a formula category like "Physics", then pick a subcategory and specific formula. Fill in the inputs and click Solve.

Note: You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: $a + 2 * (\text{SIN}(20))^3$

2.

Python calculator Settings Saved Results Modes ▾

$A = \frac{1}{2} \times \text{base} \times \text{height} =$
100.0

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

10	Dynamic inputs	Achieved	The dropdown options and text inputs will dynamically change based on previous choices (Formula mode only)
----	----------------	----------	--

1.

Python calculator Settings Saved Results Modes ▾

Formula Mode

Please select a category to view the available formulas.

Areas and Perimeters Trigonometry Polynomials Physics

Select a formula subcategory: ✓ Please choose a subcategory
Triangles
Quadrilaterals
Circles

Select a formula: Choose a formula ▾

How to Use: Choose a formula category like "Physics", then pick a subcategory and specific formula. Fill in the inputs and click Solve.

Note: You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: $a + 2 * (\text{SIN}(20))^3$

2.

Python calculator Settings Saved Results Modes ▾

Formula Mode

Please select a category to view the available formulas.

Areas and Perimeters Trigonometry Polynomials Physics

Select a formula subcategory: Triangles ▾ Select a formula: ✓ Choose a formula
Right angle Triangles
Non-right angle Triangles

Solve

How to Use: Choose a formula category like "Physics", then pick a subcategory and specific formula. Fill in the inputs and click Solve.

Note: You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: $a + 2 * (\text{SIN}(20))^3$

3.

Formula Mode

Please select a category to view the available formulas.

Areas and Perimeters Trigonometry Polynomials Physics

Select a formula subcategory: Select a formula:

Quadratics
Cubics
Quartics

How to Use: Choose a formula category like "Physics", then pick a subcategory and specific formula. Fill in the inputs and click Solve.

Note: You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: $a + 2 * (\text{SIN}(20))^3$

4.

Formula Mode

Please select a category to view the available formulas.

Areas and Perimeters Trigonometry Polynomials Physics

Select a formula subcategory: Select a formula:

Solve

How to Use: Choose a formula category like "Physics", then pick a subcategory and specific formula. Fill in the inputs and click Solve.

Note: You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: $a + 2 * (\text{SIN}(20))^3$

11	Settings mode	Achieved	The user can change the number of significant digits of the displayed results
----	---------------	----------	---

1.

Python calculator Settings Saved Results Modes ▾

Settings

Set Global Rounding Value

D.P.:
5

Set Rounding

How to Use: Input a number from 1-10 to round all answers in formula mode and calculation mode to that number of decimal places.

2.

Python calculator Settings Saved Results Modes ▾

Settings

Set Global Rounding Value

D.P.:
2

Set Rounding

Rounding set to 2 decimal places.

How to Use: Input a number from 1-10 to round all answers in formula mode and calculation mode to that number of decimal places.

3.

Python calculator Settings Saved Results Modes ▾

1/3 =
0.33

a = 0 Save to Variable

Keyword to save result

Save to Table

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

12	Regression mode functionality	Achieved	The user can successfully input a table of values and receive a graph, a regression equation and the R^2 value
The test variables should result in the equation $y = 2x + 0$ with $R^2 = 1$ The graph should be a straight diagonal line upwards that passes through all plotted points 1.			

Regression Mode

x	y
x0	y0
x1	y1
x2	y2
x3	y3
x4	y4

Add Row Remove Row Calculate Regression

How to Use: Fill in the table of X and Y inputs. Click Add row or Remove row as needed. Once ready, click Calculate regression.

2.

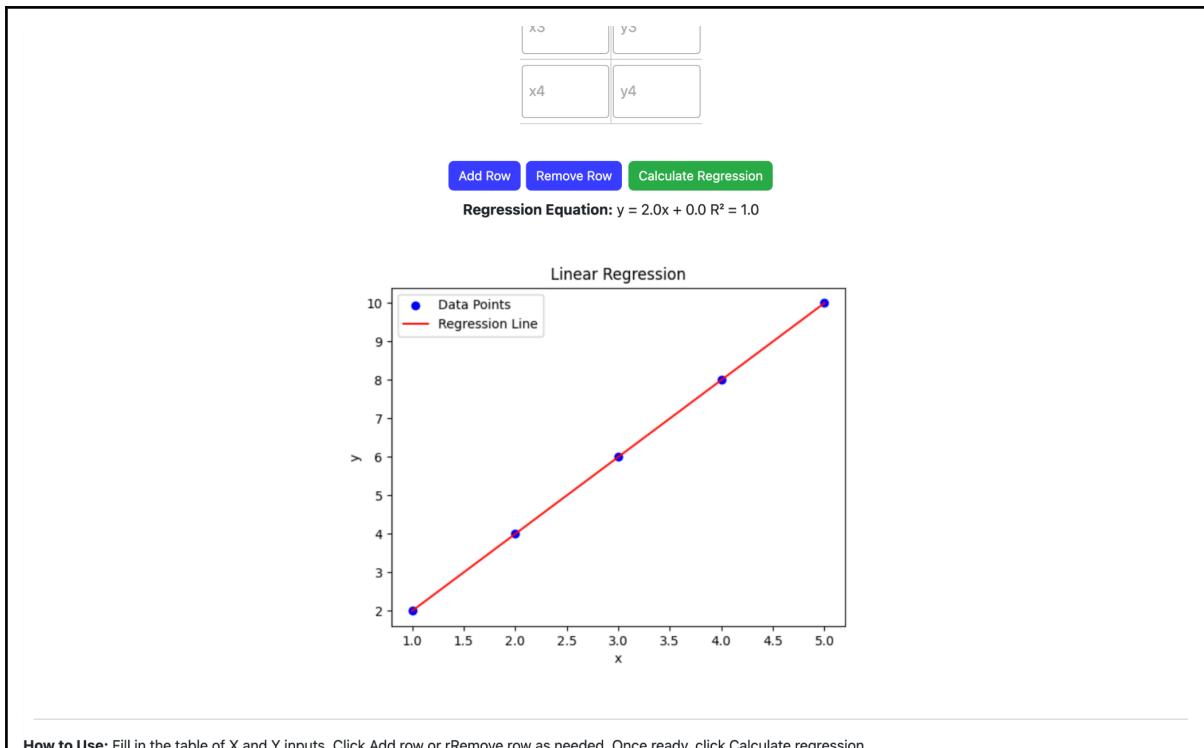
Regression Mode

x	y
x0	y0
1	2
x1	y1
2	4
x2	y2
3	6
x3	y3
4	8
x4	y4
5	10

Add Row Remove Row Calculate Regression

How to Use: Fill in the table of X and Y inputs. Click Add row or Remove row as needed. Once ready, click Calculate regression.

3.



How to Use: Fill in the table of X and Y inputs. Click Add row or Remove row as needed. Once ready, click Calculate regression.

13	Regression mode functionality (i)	Achieved	The user can add or remove rows from the input table
----	-----------------------------------	----------	--

1.

Python calculator Settings Saved Results Modes ▾

Regression Mode

x	y
x0	y0
x1	y1
x2	y2
x3	y3
x4	y4

How to Use: Fill in the table of X and Y inputs. Click Add row or Remove row as needed. Once ready, click Calculate regression.

2. Adding a row

Regression Mode

x	y
x0	y0
x1	y1
x2	y2
x3	y3
x4	y4
x5	y5

[Add Row](#) [Remove Row](#) [Calculate Regression](#)
3. Removing a row**Regression Mode**

x	y
x0	y0
x1	y1
x2	y2
x3	y3

[Add Row](#) [Remove Row](#) [Calculate Regression](#)

How to Use: Fill in the table of X and Y inputs. Click Add row or Remove row as needed. Once ready, click Calculate regression.

14	Math function substitution	Achieved	Python math modules can be input in the input field of calculation mode and resolved
----	----------------------------	----------	--

The test function will be the square root of 64, which should result in an answer of 8
1.

Calculation Mode

SQRT(64)

Submit

How to Use: Input your equation and click calculate to calculate. You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: a + 2 * (SIN(20))^3

2.

SQRT(64) =

8.0

a = 0 ▾ Save to Variable

Keyword to save result

Save to Table

How to Use: Save the result permanently by entering a keyword and clicking Save to Table, or choose a letter from the dropdown to save as a variable while using the website.

15	XSS and SQL injection protection	Achieved	JavaScript or SQL queries won't run when input into text inputs
----	----------------------------------	----------	---

Testing for XSS: <script>alert("XSS Detected")</script> if vulnerable a system alert saying "XSS Detected" will be displayed

1.

Calculation Mode

<script>alert("XSS Detected")</script>

Submit

How to Use: Input your equation and click calculate to calculate. You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: a + 2 * (SIN(20))³

2.

Calculation Mode

Submit

Invalid characters detected. Please avoid using <, >, ', " or ;.

How to Use: Input your equation and click calculate to calculate. You can also input the keywords of saved results or variable names too!

Math functions: SIN, COS, TAN, SQRT, PI, CBRT, EXP, LOG, DEGREES, RADIANS, ACOS, ASIN, ATAN, E, FACTORIAL

For example: a + 2 * (SIN(20))³

Testing for SQL injection: Test5 OR 1=1, if vulnerable, the saved result Test5 would be displayed

1.

Saved Results

Sort by: Keyword Sort

Search by keyword:

Test5 OR 1=1

Search

Keyword	Result	Action
Test1	20.0	Remove
Test2	30.0	Remove
Test4	72.0	Remove
Test5	100.0	Remove
Test8	50.0	Remove

How to Use: Enter a keyword to search for a specific result, or use the dropdown to sort the results. Click remove to remove a result.

2.

Saved Results

Sort by: Keyword Sort

Search by keyword:

Test5 OR 1=1

Search

Keyword	Result	Action
No saved result found.		

How to Use: Enter a keyword to search for a specific result, or use the dropdown to sort the results. Click remove to remove a result.

16	Invalid redirect protection	Achieved	Inputting links to malicious files will not redirect the user to said HTML files.
----	-----------------------------	----------	---

Test URL: <http://localhost:4000/?next=redirected.html> &

<http://localhost:4000/redirected.html>

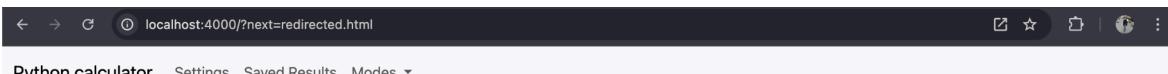
If vulnerable, it should redirect to the empty test HTML file redirected.html

1.



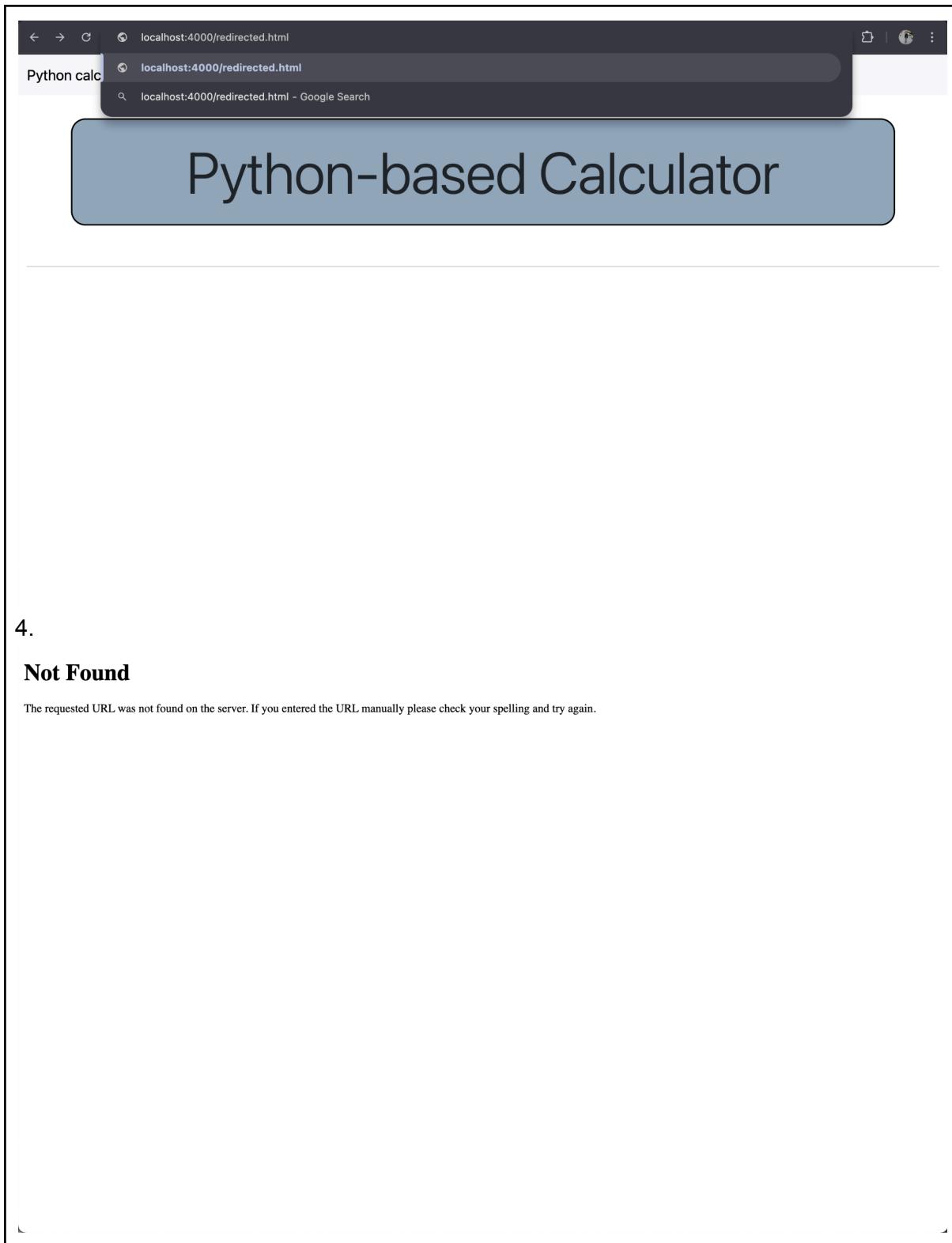
Python-based Calculator

2.



Python-based Calculator

3.



4.

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.