




CS346 ASSIGNMENT 2

EDUCATIONAL SOFTWARE FOR CHILDREN

Prepared By :

Group 5A

1. Himangshu Deka(210101050)
 2. Himanshi Gautam(210101051)
 3. Janapati Sai Ajay(210101052)
 4. Jethwani Siddhant Pratap(210101053)
 5. Kannan Rustagi(210101054)
 6. Kella Usha Shree(210101055)
- 

Problem Structure

We have to create educational software to teach students fundamental concepts while making learning fun and interesting.

The software should provide the following functionalities-

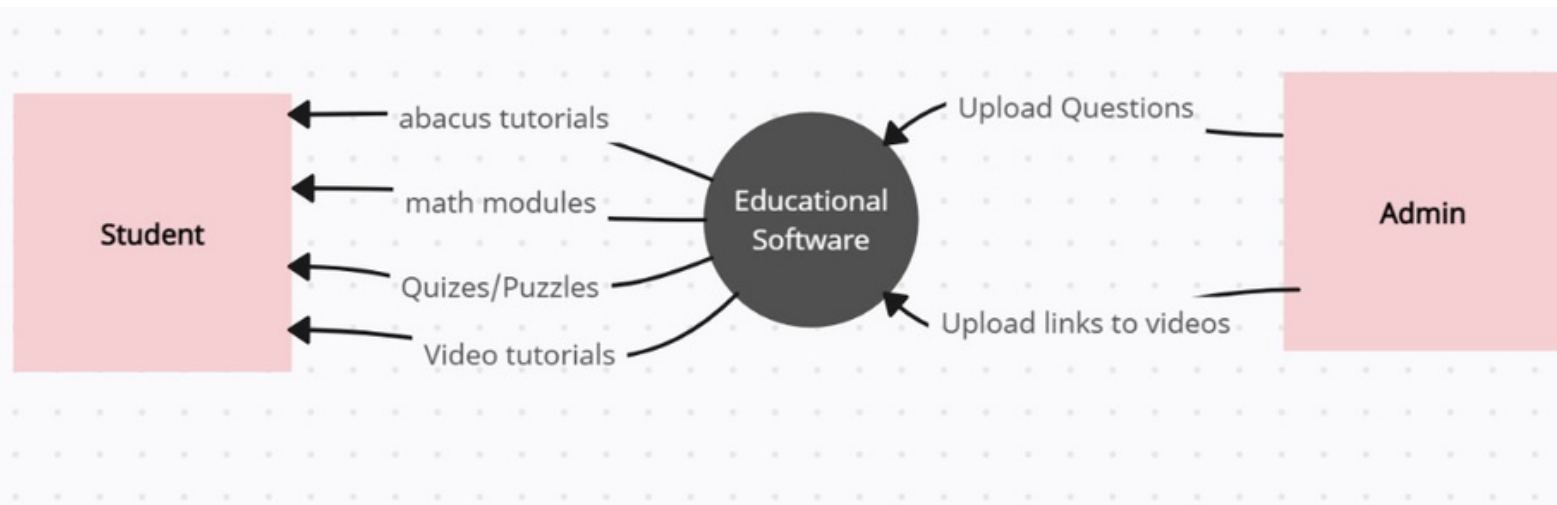
1. Teaching arithmetic operations- addition, subtraction, multiplication, and division using children-friendly visualization.
2. Teaching some arithmetic operations using abacus.
3. Making YouTube video tutorials available for quick learning.
4. The users should be able to attempt quizzes and puzzles on different topics and get feedback.



Data Flow Diagram for the proposed Solution



Level 0 DFD

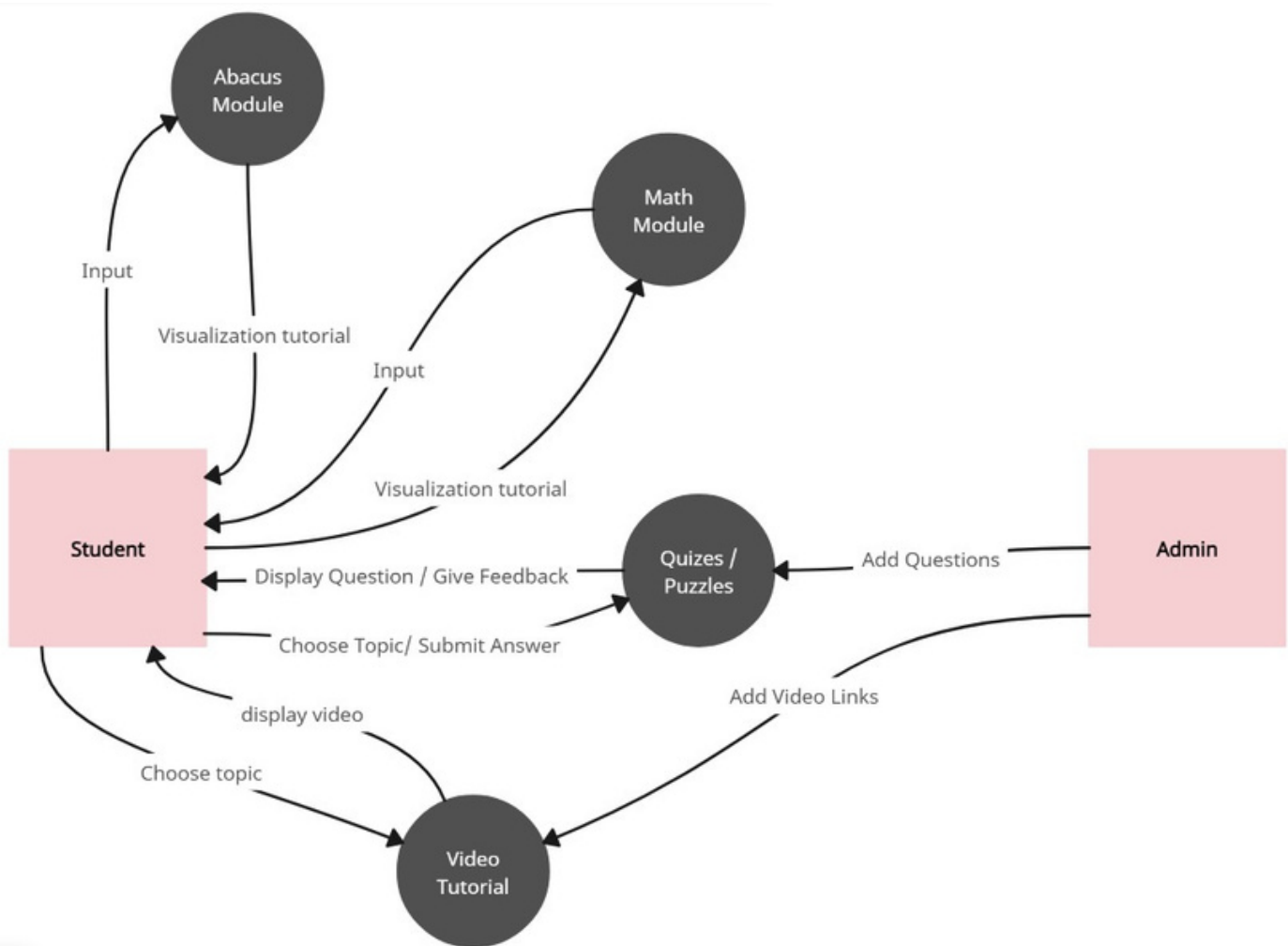


There are 2 external entities in the context DFD-

1. User- students are the stakeholders for whom the software is intended
2. Admin- the function of this entity is to manage the database.


It represents the educational software as a single process and shows the system boundaries, external entities and how they are interacting with the software.

Level 1 DFD



Now we break down the single process into several main sub-processes. It consists of the following sub-processes-


1. **Math Module-** The student will provide input to this process, i.e., the operands and the operation to be performed. Then the process shall show how the operation is occurring on the operands provided using children-friendly visualization and then the output will be displayed to the user.



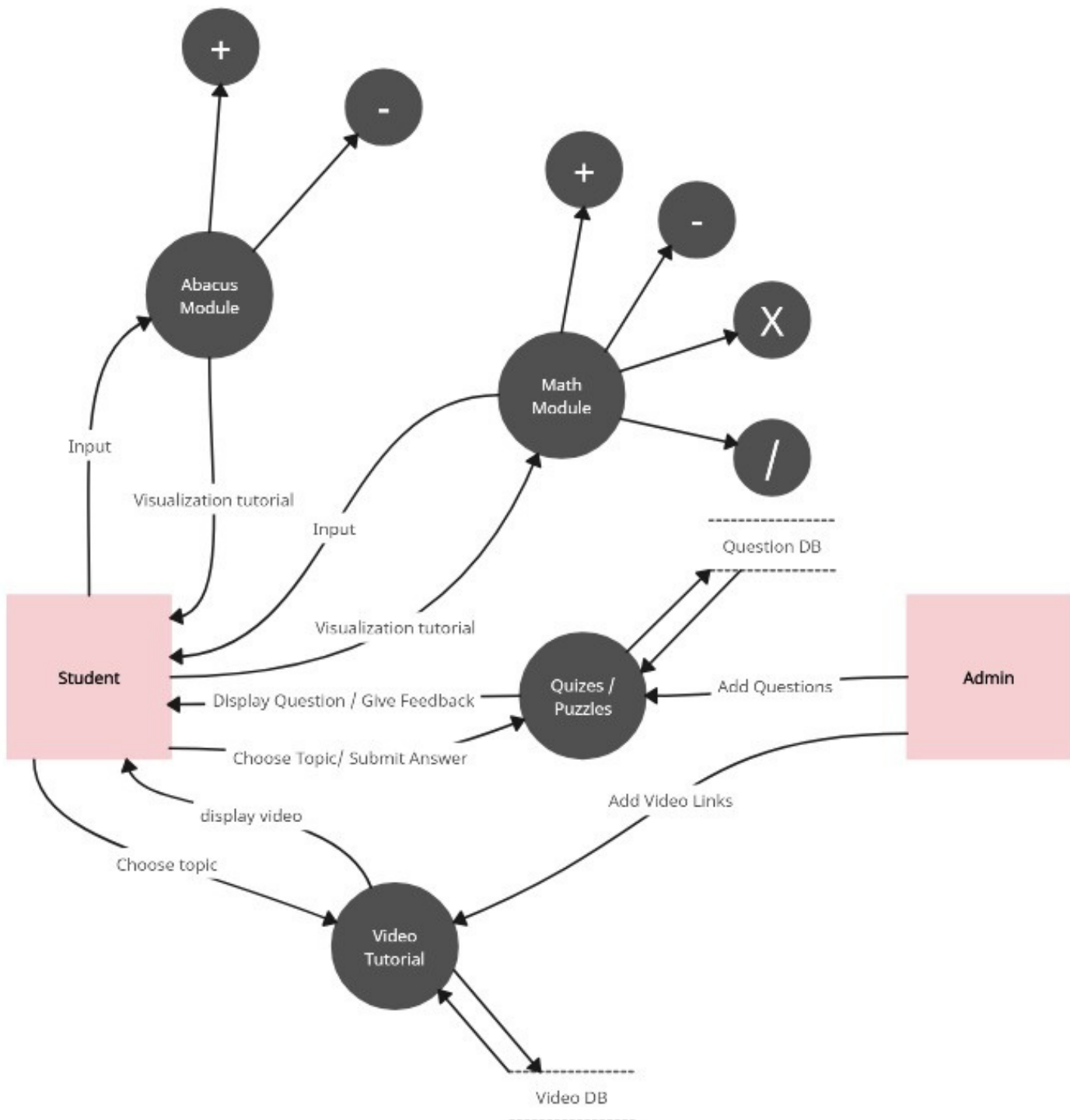
2. **Abacus Module-** This module will accept as input the operation to be performed and the operands and then will show using effective visualization how the operation is occurring using abacus techniques and then finally display the output.


3. **Quizzes/Puzzles Module-** This process will accept input as the topic on which the student wants to be questioned and then display questions that the student will have to answer. The process will then provide feedback.

4. **Video Tutorial Module-** This process will accept input as the topic on which the student wants to view a video tutorial and then it will show a list of videos available on that topic. The student can choose to play any of the videos.



Level 2 DFD





Each Sub-Process is now broken down into its specific functionality :

Abacus Module

- Addition
- Subtraction

Math Module

- Addition
- Subtraction
- Multiplication
- Division

Quizzes/Puzzle Module

- DB connection for question retrieval

Video Module

- DB connection for video link retrieval
- 

MODEL FOLLOWED

The waterfall model consists of the following steps:

1. Resource Gathering and Analysis of Problem Statement
2. Design
3. Development
4. Testing
5. Deployment
6. Maintenance

For this project, we are following the **waterfall model**.

In the first 2 steps, The problem statement analysis is performed and possible solutions have been proposed and finally compiled into a brief report.

For the 3rd Part, development of the proposed solution will take place using Visual Basic as the programming language and using Visual Studio as the platform while using MySQL for database and ODBC for database connection.

For the 4th Phase i.e. Testing Phase, our team will test out all the features and look out for bugs, if any are found, they will be fixed.

Setting up the Database

- We will use a MySQL database in our software to store the appropriate relations as required. We can use a tool like phpMyAdmin or MySQL Workbench to create a new database.
- We will then set up an ODBC connection to the MySQL database we have created. This involves configuring a Data Source Name (DSN) using the ODBC Data Source Administrator tool present on the system.
- Once we're done setting up the database, we'll use it in different modules of our software like the video tutorial module and quizzes/puzzles module to store the appropriate relations as required.
- **Video Tutorial Module:** We will use the database to store a table that contains a set of relevant YouTube links containing educational content on various topics. The table will contain 2 columns- topic, and video link.
- **Quizzes/Puzzles Module:** We will use the database to store a table that contains a set of questions on various topics. The table will contain 3 columns- topic, question, correct answer.

MATH MODULE

This feature helps the children learn basic mathematical implementations like addition, subtraction, multiplication, and division and it also makes the learning process interesting by using visuals like tally marks.

Scope of this Module-

There will be 4 functionalities in basic math module-

- Addition
- Subtraction
- Multiplication
- Division

Each functionality is shown by one button. When a user wants to perform a particular operation they click on the respective button to perform that operation. Users can give the inputs only after selecting the operation they want to perform.

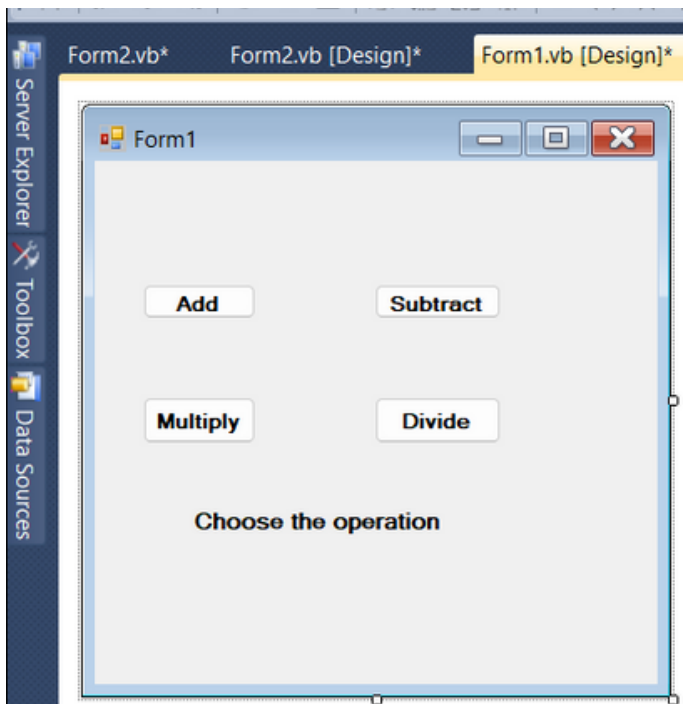
Working :

1. User is to select the operation they want to perform using the buttons provided.
2. After selecting the operation a text will be shown to them asking them to give input.
3. Input of two numbers is given using text boxes.
4. Once the input is provided the software clicks a button to perform the operation. Once the button is pressed it checks whether the provided input is valid or not.
5. After checking the validation the respective operation is performed if the input is valid or a text will be displayed showing invalid input.
6. Once the result is displayed there will be another button “Next” by clicking on it we will display the process of how we got the result.
7. The software uses visual aids such as tally marks or sticks and colorful illustrations to help children visualize the mathematical concepts. Picture box can be used to display the visualizations.
8. Once the process is completed they can either repeat the process (by clicking on add button again) or can clear the input and visualize the same operation using different inputs (using clear button) or can go back (using exit button) and change the operation.
9. While performing division the output values show remainder and quotient.

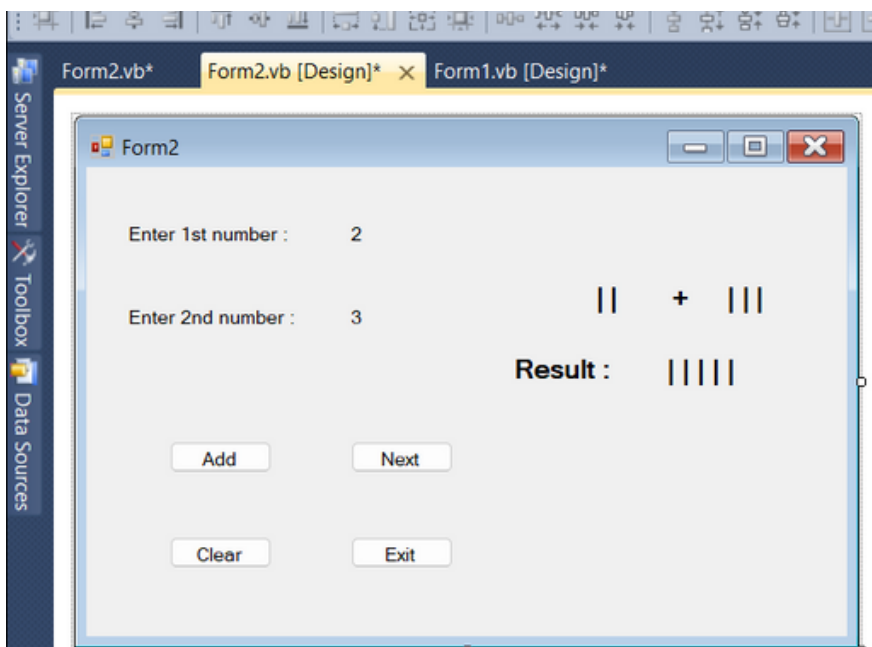
Boundary Cases :

1. Only whole numbers are used for addition, multiplication and subtraction.
2. 0 will be considered invalid input if given as divisor in division.
3. In subtraction, the 1st number should be greater than the 2nd number.

Examples :



The same way we will also show the visualizations for subtraction, multiplication and division.



ABACUS MODULE

The objective of this module:

1. Aid the stakeholders in achieving proficiency in basic mathematics using an abacus.
2. Improve upon the design goal of part (i) by improvising support for addition and subtraction to the abacus that shall provide for better visualization, with possible further extension for multiplication and division

Scope of this module:

1. Provision of 8 columns of 5 beads each, with each column being groups of 1 and 4 beads respectively this mimics our 8-column abacus(i.e. 8 digit supportive abacus).
2. The abacus can be implemented in code as a 2D Array with the required number of rows and columns (i.e. 8×5)
3. Visually, Each column can be a Line Element, and Each bead shall in turn be possibly a VS Basic Form button / Oval element that shall move down/up, as the rules of the abacus dictate, with proper handling done.
4. Each bead can have 2 states and an associated weight
 - State 1: Bead Inactive
 - State 2: Bead Active
 - Weight denotes the weight of the bead that is added to the place value when the bead is in State 2.
5. The buttons shall be clickable while maintaining the properties of abacus arithmetic i.e. In the lower stack of any column, the lower beads cannot be accessed without touching the upper beads i.e. the lower beads can only transition from State2 -> State 1 when the beads above them are in State1, and the Upper Beads can only transition from State1 -> State 2 when the beads below them are in State2.

6. An Option picker will be present for selecting the operation to be performed and Two buttons "Enter Input" shall be placed next to both the abacus, which will be used to take input from the user of the numbers, that shall be used for the actual calculation.
7. The user shall click on the 'bead' buttons, with a valid click resulting in the floating of the buttons and updation of the abacus arrays.
8. As a column in the abacus represents a decimal number, we shall use 2 8 length arrays for the abaci, initialised as all 0's
9. When Proceed to Calculation is pressed, the calculation starts and all the buttons and beads stay disabled until calculation finishes.
10. In the logic for respective calculation, after each iteration of array arithmetic is done, the corresponding abaci column is updated thus providing us with the visualisation of the arithmetic.
11. Upon completion of all iterations of calculation all buttons and beads are enabled.
12. A provision of a 'Reset' button shall also be maintained that shall clear all the inputs, interrupt the calculation iterations, set both the arrays to all 0's and enable remaini.

Limitations

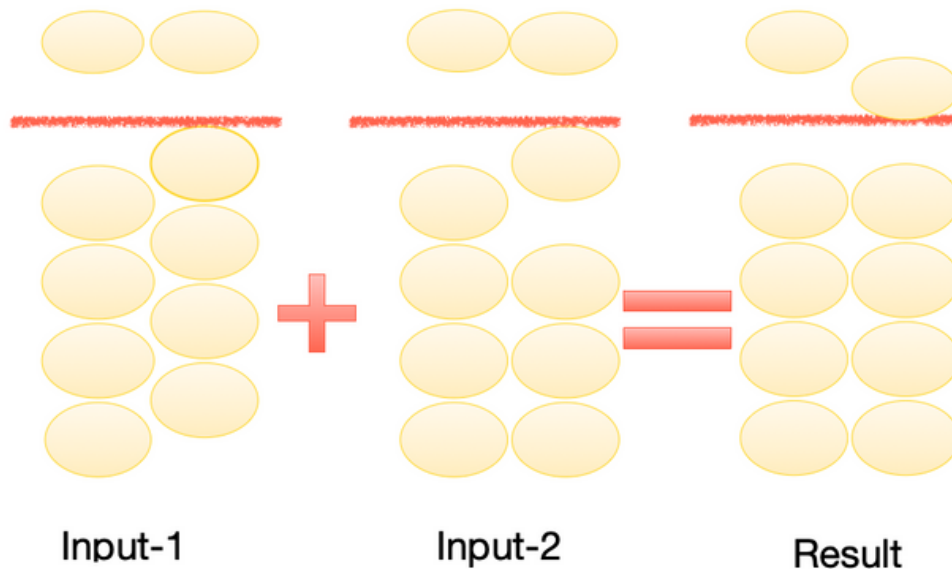
- a. Our abacus module is currently planned for functioning of addition and subtraction of positive integers, bearing atmost 8 digits.
- b. In the subtraction part, the first input must be greater than the second one else a message prompting user to do the same is shown.
- c. If the result of addition crosses 8 digits then a user prompt is generated that overflow is generated.

Corner Cases and Exception Handling

1. Appropriate carry-outs must be taken care of if addition is considered.
2. If data overflows, we display a warning saying there has been an overflow.
3. While considering subtraction, borrows must be taken care of.
4. If the first number is smaller than the second number warning is issued, our programme deals with only positive results

Design Visualisation

- In the left column ,the abacus represents the first operand and the second column represents the second operand.
- Here we are providing a sample visualisation of how the addition (and subtraction would be visualised



Case-1:

Case-1:

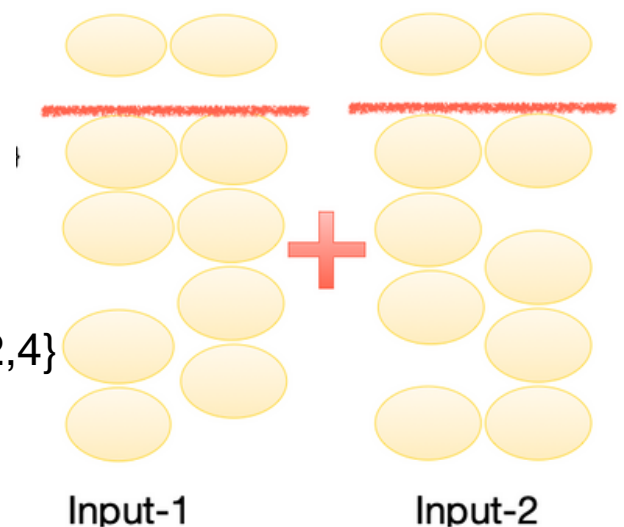
(For a 1 digit number :The addition or subtraction is a single step process and we have only one step in our visualisation)

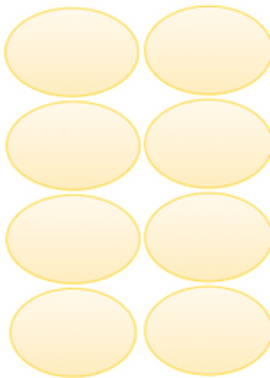
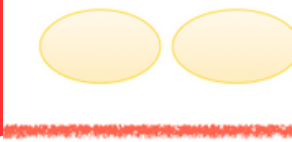
Here we are adding 4 to 1 and our result is 5,as represented by the upper bead in the resultant last colum of resultant abacus

Case-2:

Our inputs have more than one digits:24+31

The input vectors would be as {0,0,0,0,0,0,2,4} and {0,0,0,0,0,0,3,1} now upon iterating we think about providing the visualisation as follows:





Iteration-1

Step-1::

We initialise our resultant vector as $\{0,0,0,0,0,0,0,0,0\}$ and thus our answer abacus is set to all 0's.

Step-2:

After the first iteration, the unit digits are added and appropriate carry outs (if any) are taken care of, our result abacus is as shown and the answer array is $\{0,0,0,0,0,0,0,0,5\}$.

Step-3:

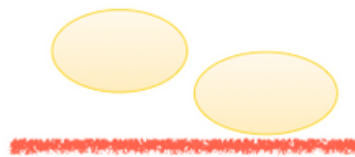
After the second iteration the tens digit is taken care of and our result abacus is as shown and the result array is $\{0,0,0,0,0,0,0,5,5\}$

Step-4 to 8:

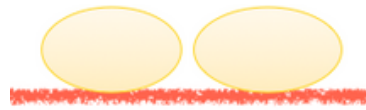
Similarly, the iterations of addition proceed in this fashion and we get the final answer, as we are only showing 2-digit addition as a specimen, hence we are omitting it in this report.

****At the end of all these 8 iterations we would have the final result abacus, as shown in Iteration-3 to 8.**

****Note: Visualisation of subtraction follows similarly and shall follow the same pattern, hence we are omitting it here to avoid repetition.)**



Iteration-2



Iteration-(3-8)

Further possible extensions subject to availability of time and other constraints:

Support for multiplication and division could also be added, if

1. Time permits

2. Addition subscribes to the limitation as documented here

3. If it doesn't, then depending upon provision of subsuming new documentation at the time of finalising the software decision shall be taken.

VIDEO TUTORIAL

MODULE

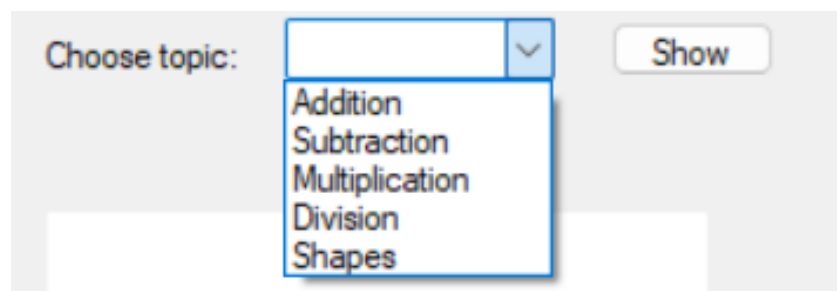
1. The interface for this feature will contain a ComboBox containing the list of topics we have videos available on then the user can select one.

2. After selecting the topic and clicking on the 'show' button, all the videos available on that topic will be fetched from the corresponding table in the database and shown on separate web browser components on the screen. Then the user can play the one they want on the window.

3. Inside the event handler for the show button, we will establish a connection to the MySQL database using the ODBC connection string and write a SQL query to select the YouTube links from the database table. Then execute the query using an OdbcCommand object and retrieve the results using an OdbcDataReader.

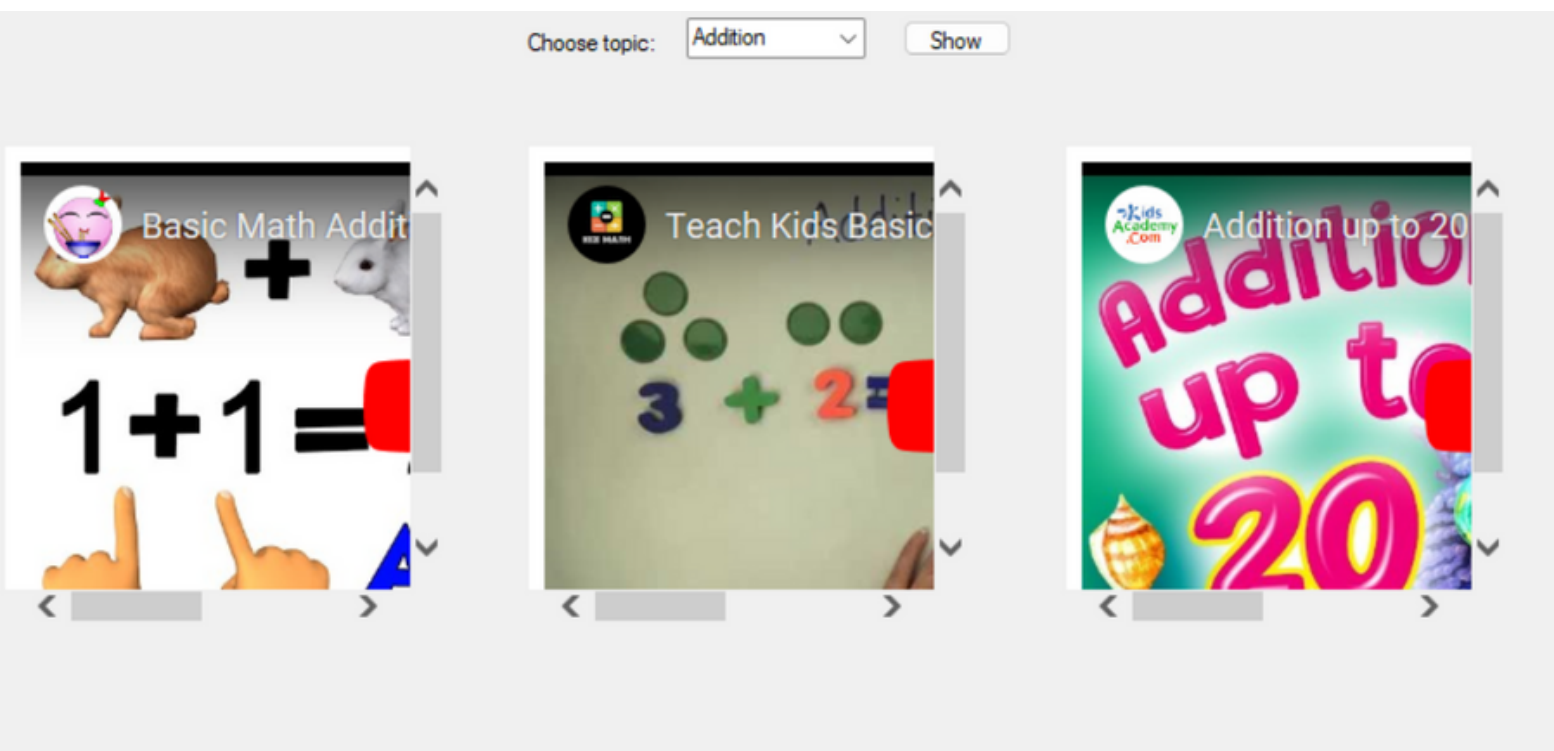
Thus, the user can play any video from the list of available videos added for reference in the software.

INTERFACE



Showing the dropdown of topic choices

How the content will be visible on the screen



QUIZZES/PUZZLES

MODULE

Mathematics Quizzes:

1. We can create some predefined questions on various topics: addition, subtraction, multiplication, and division, and store them in the database.
2. Whenever a user wants to attempt a quiz, we can randomly select a fixed number of questions(say, 10) and display them to the user to solve. Randomizing the order of questions in quizzes for kids fosters engagement, prevents memorization, and promotes critical thinking while accommodating diverse learning styles and improving long-term retention of information.
3. Once the user is finished attempting the question, they'll be notified if they answered it correctly or not and at the end of the quiz, all the correct answers will be displayed, along with their answers so that they can learn from their mistakes.
4. We can also gamify our quizzes by including credits/points for answering correctly so that the kids are motivated to attempt the quizzes and test their knowledge.

Example:

Before the user enters the input

1.  +  = 

If the entered input is correct

1.  +  = 


If the entered input is incorrect

1.  +  = 



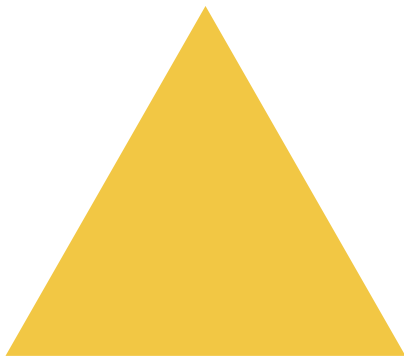
Shape and Pattern Puzzles:

There are various types of puzzles that we can include in our software like:

1. **Identifying Shapes:** Develop puzzles where students are presented with a shape and must identify its name or properties. Display various shapes (e.g., square, circle, triangle) and ask the student to select the correct name from multiple-choice options.
 2. **Completing Patterns:** Create puzzles where students must complete a pattern by selecting the correct shape or number to fill in the missing piece.
 3. **Sorting Shapes:** Implement a sorting activity where students drag and drop shapes into the correct categories based on their attributes (e.g., color, number of sides).
- 

Example:

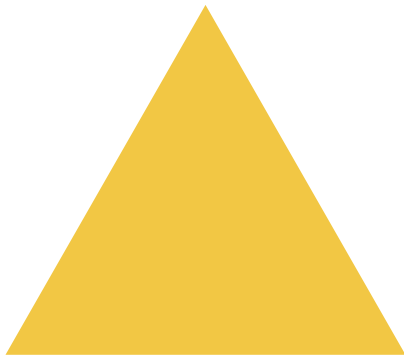
Before the user enters the input



What shape am I?

|

If the entered input is correct



What shape am I?

Triangle

If the entered input is incorrect



What shape am I?

Circle



THANK YOU!

