University of Dhaka

Department of Computer Science & Engineering

Project Report:

Fundamentals of Programming Lab (CSE-1211)

Project Name:

Mechanical Calculator

Team Members:

1. *Md. Ehsan Uddoula Pahlowan (Roll-50)*
2. *Kanij Fatema (Roll-18)*
3. *Md Shariful Islam Rayhan (Roll-41)*

Introduction

Physics Mechanical Calculator is an SDL based project written in C and C++ language. The code of this project is simple, clear, and easily customizable. Based on some special mechanical topics of physics, our team has implemented this useful project. Using this , students can calculate the problems as well as visualize them through animations. Also they can check if their solution is correct or not. Visualization of the problems through animations has made this project distinctive and unique. It will make the study & relative topics interesting to everyone.

Objectives

Our project will be helpful for SSC/HSC level students. We implemented the calculator on the basis of physics topics. Those are very important and common topics in **board exams and university entrance tests.** Moreover the topic options are dynamic with animation to explain what is the real condition of a given problem. Students can choose their variable which one they want to find out and depending on this, we have placed the input options to take the values. Our calculator will do the calculations & show the answers with the animated case. Besides, Edtech platforms can use our calculator in their physics section if they host it on their website or in a mobile app.

Project Features

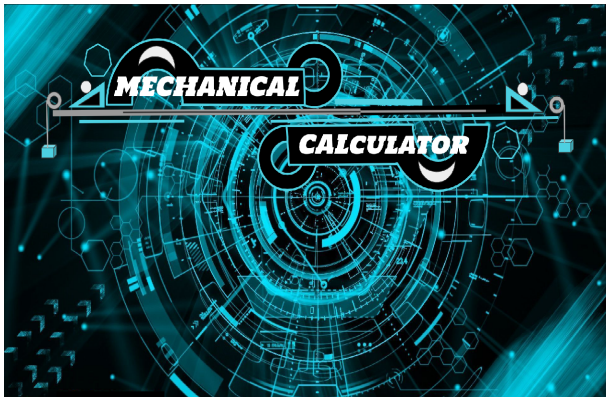
- Simple & clear code in a structured and modularized manner with the proper use of the header files & proper commenting.

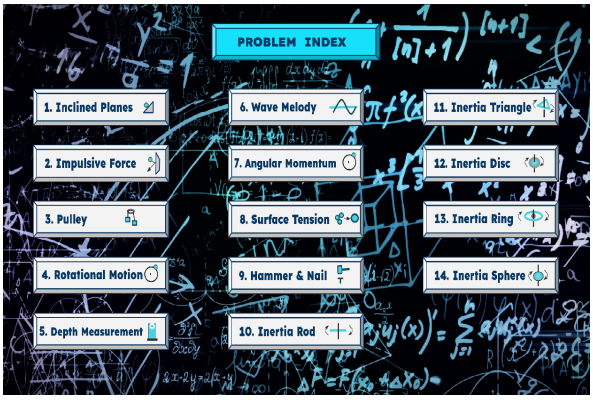
- The project contains some problem options on the menu . The user can choose which problem he wants to solve.

- The problems have their relative input & output cases inside the options.

- The photos of almost all the cases of the problems and all the animation sprites, menu buttons are made on own.

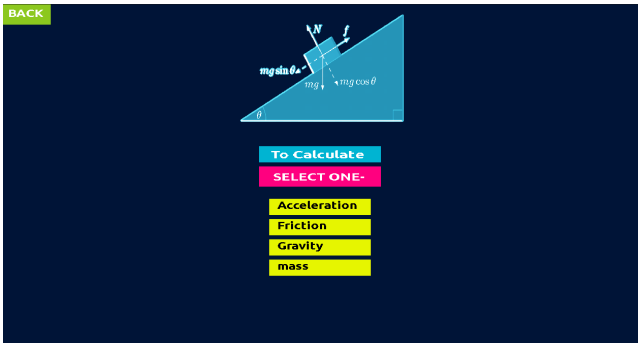
- On the starting, after the loading page , the dynamic menu page will appear.Here one will select the problem. Along with the name of the problem, each option contains a simple tiny figure of the problem to give more clarification of the problem cases.



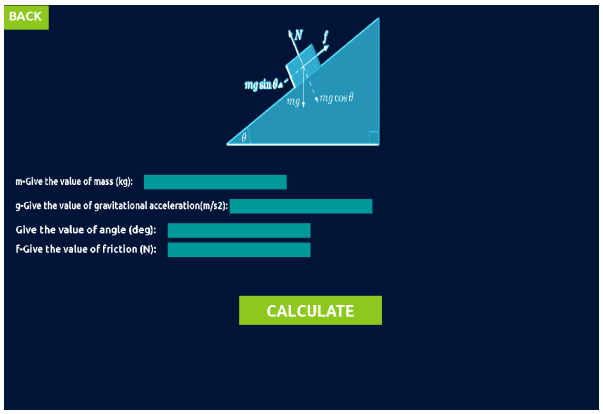


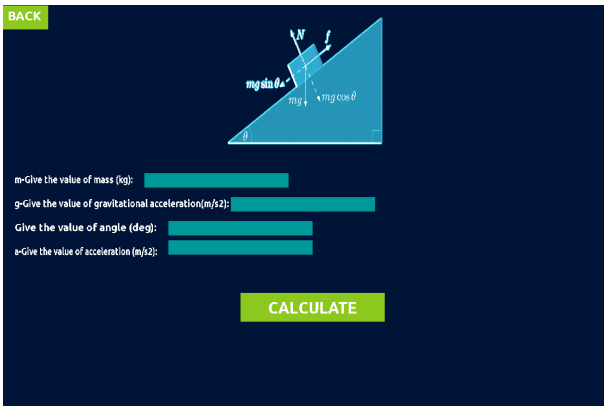
- **1. Inclined PLanes**

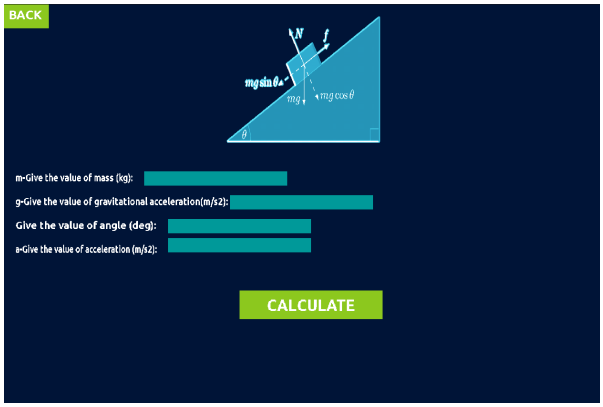
  Selection of this menu will show the below page and prompt to select an option depending on which value one wants to calculate. There are 4 cases available.

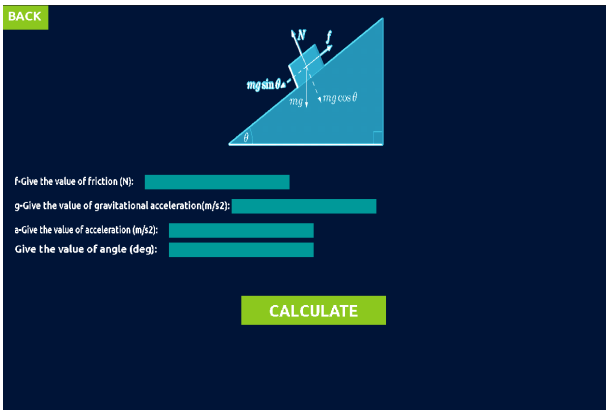


Selection of the any 4 options will open the 4 different input ways

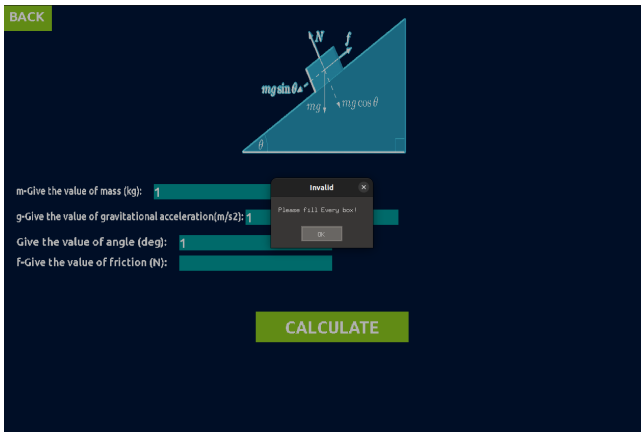


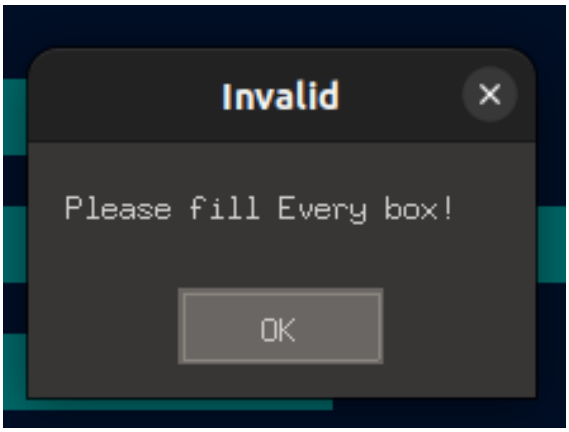




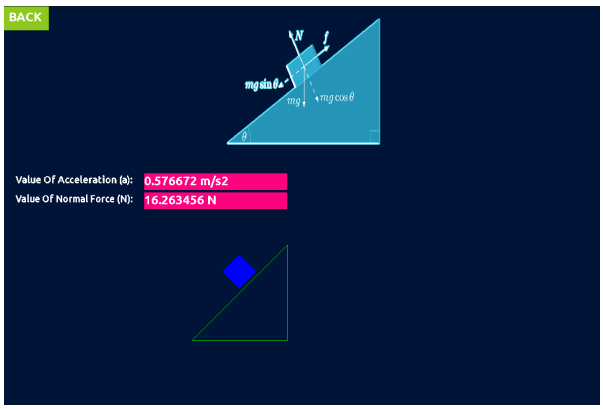


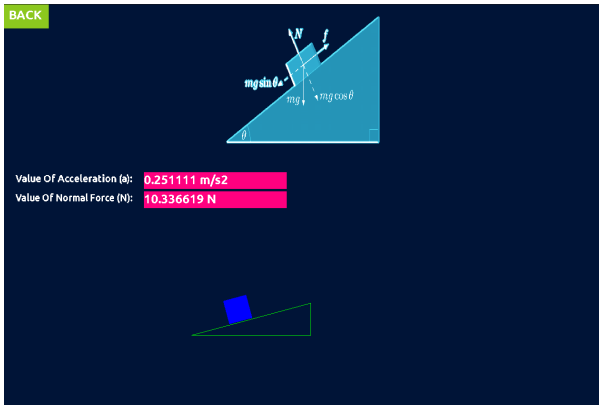
If any of the input boxes gets skipped a message box will be shown saying to fill up every input box.





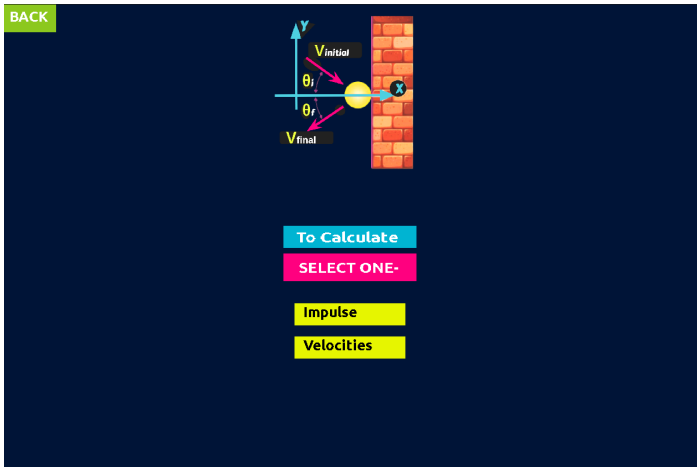
After putting the values, when the calculate button will be clicked the answer page with the animation will appear. The animation gets changed according to the input values.



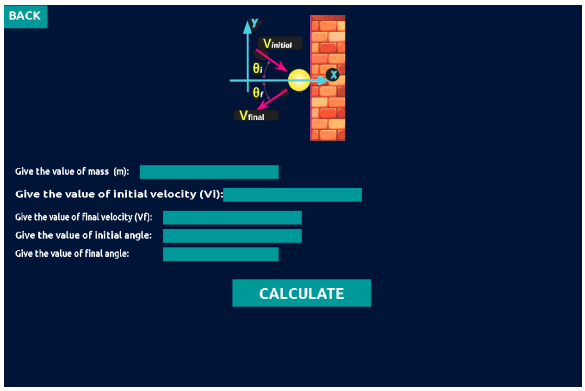


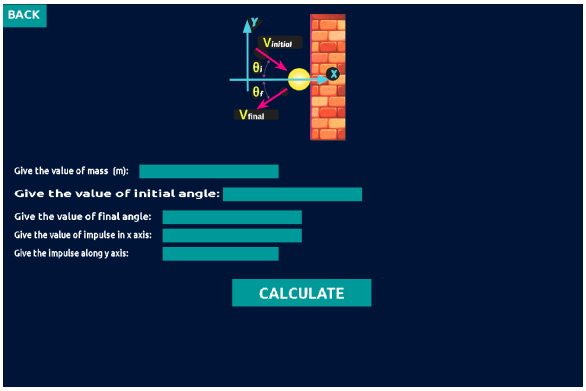
Then by pressing the back button or the quit “X” button the user can go to the back pages and calculate again or go to the menu to calculate another of the problems.

- **2. Impulsive Force**
  This option will arrive like the below image.

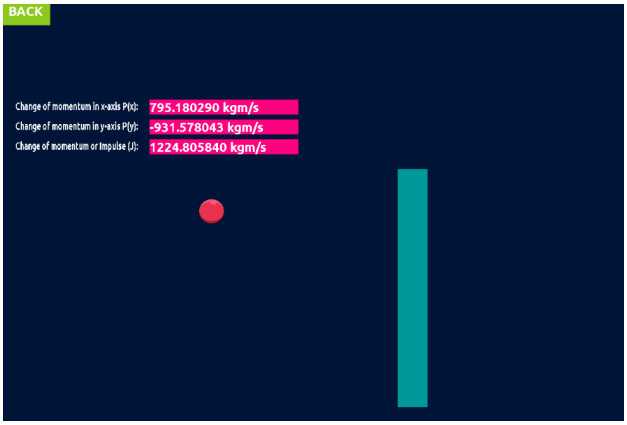


There are 2 options for getting the outputs.



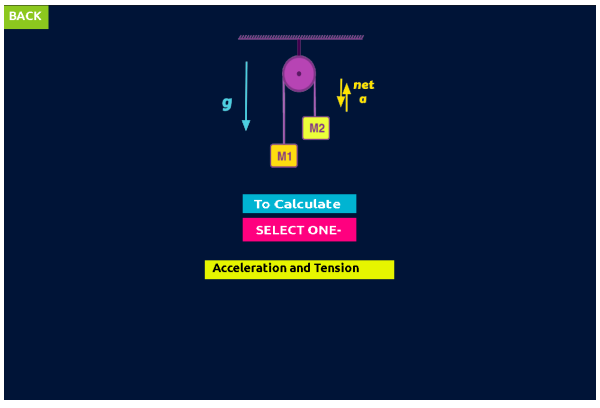


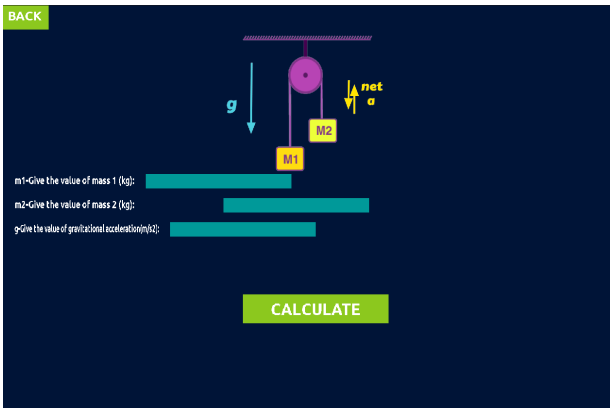
The calculation will appear with the animation of that specific problem.



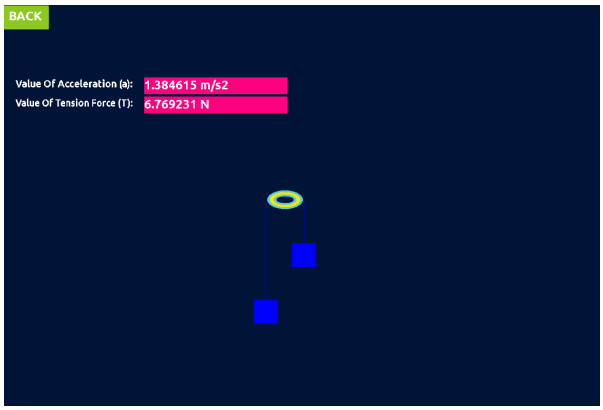
- **3. Pulley**

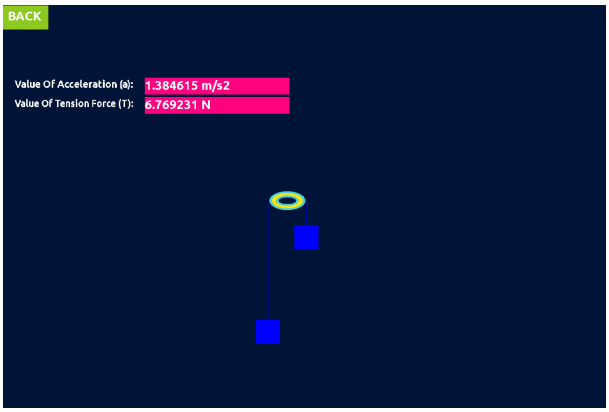
  The option clicking appearance-





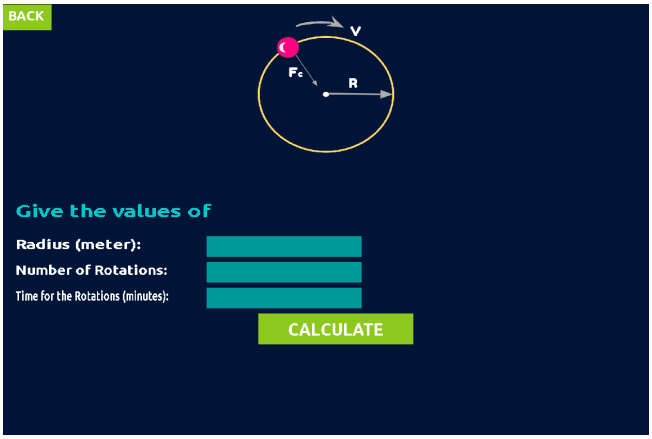
The calculation with the animation. Animation changes according to the inputs and outputs.



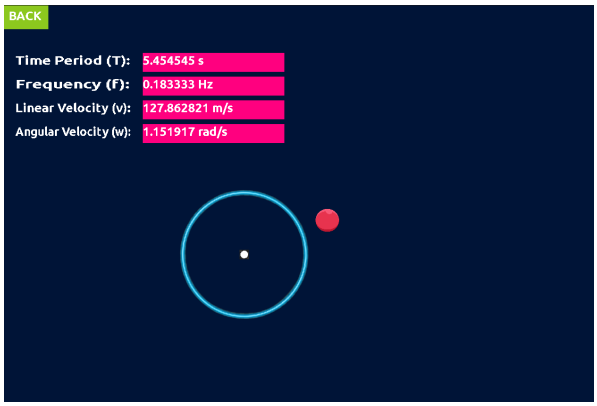


- **4. Rotational Motion**

  The problem page-



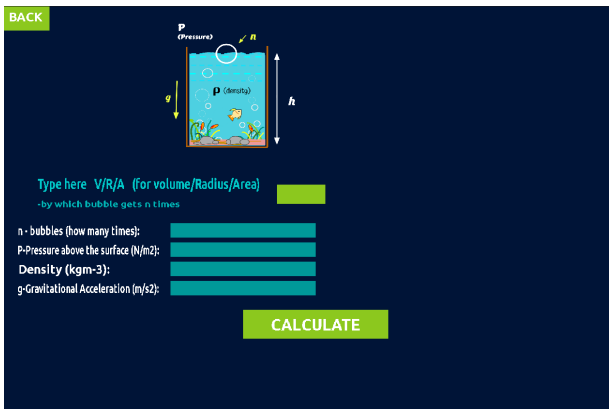
The calculation and animation page-



The radius,speed etc get changed according to the input values.

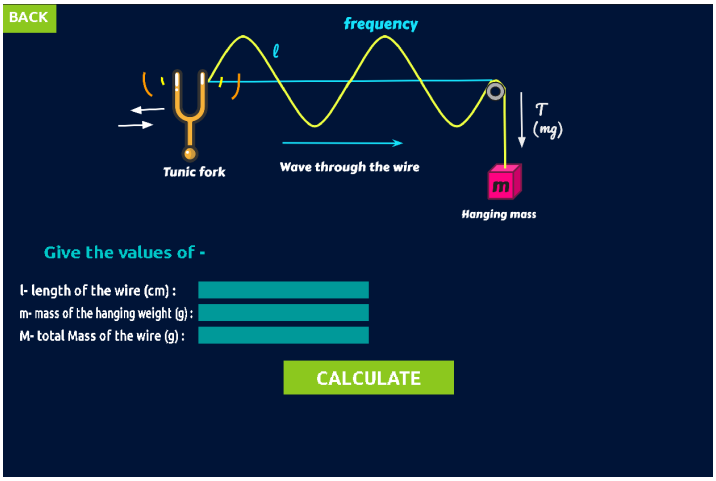
- **5. Depth Measurement**

  The problem page & the calculation, animation page-

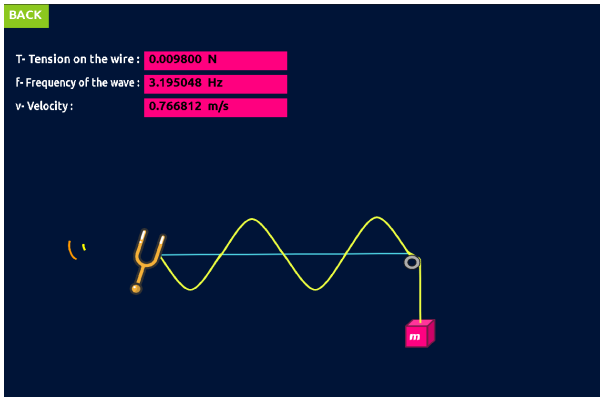


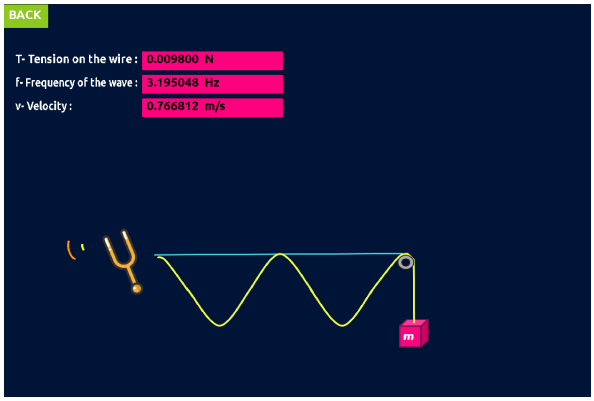
- **6. Wave Melody**

  The problem page -



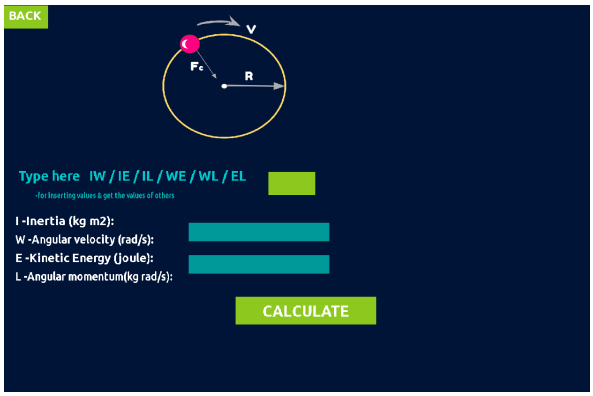
The calculation, animation page-

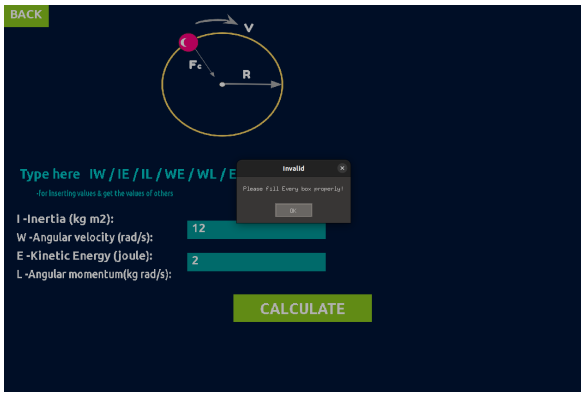


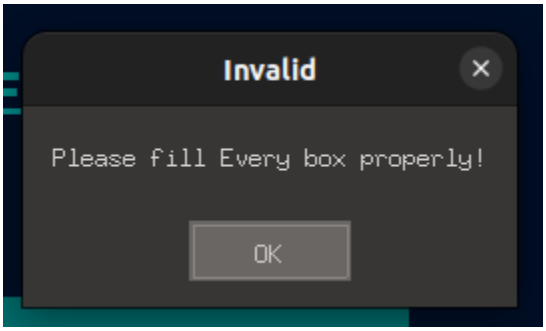


- ## 7. Angular Momentum

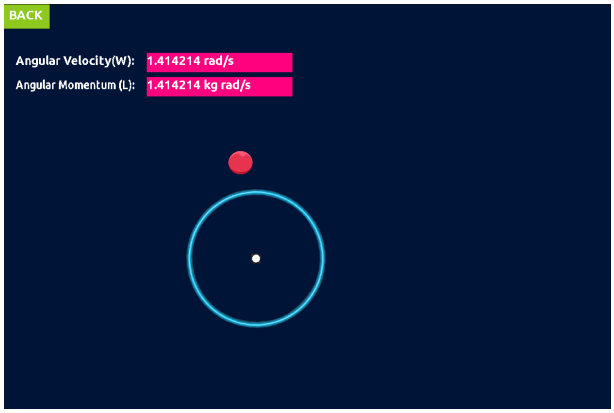
  The problem page-here are 6 different input & output cases. There will be a message box if the values/options are not given properly.

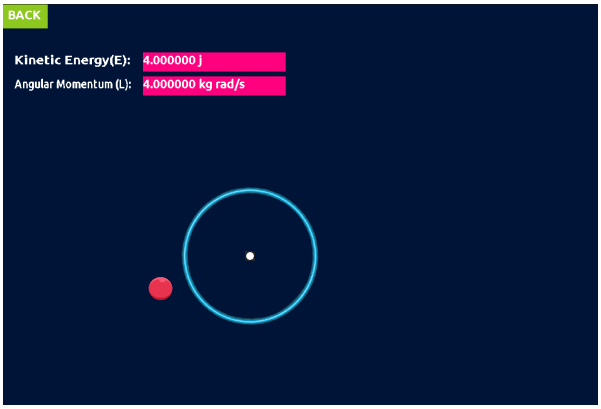


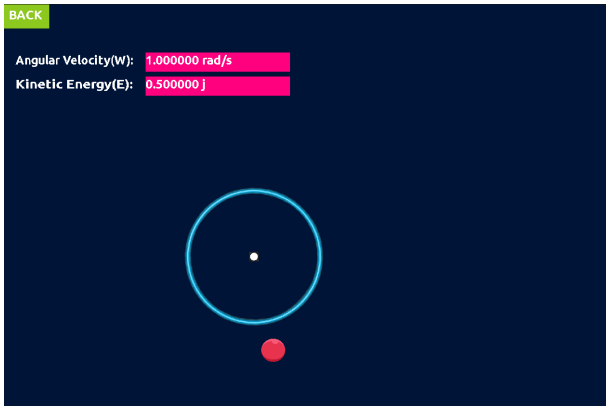


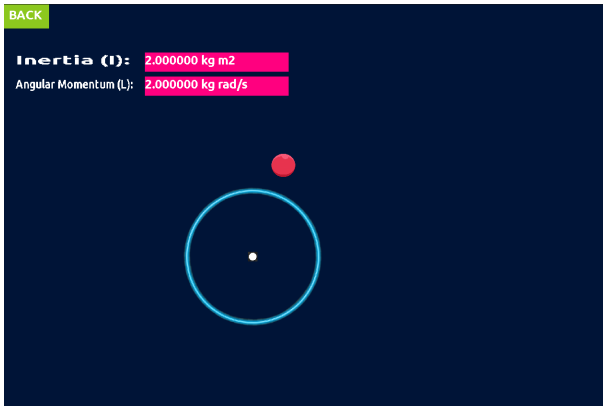


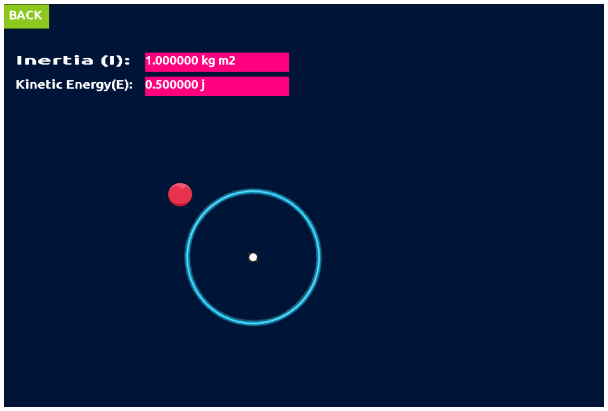
The answers of the 6 different cases are different but the animation remains the same.

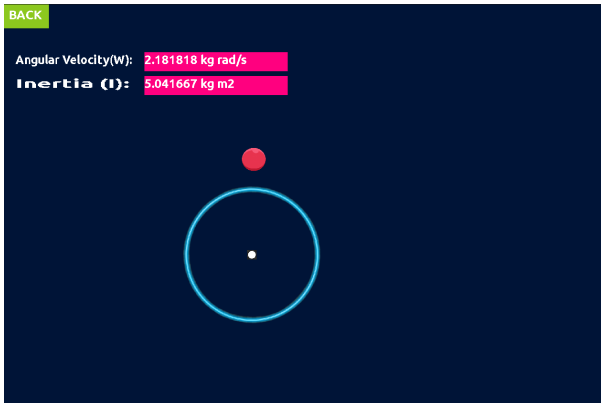




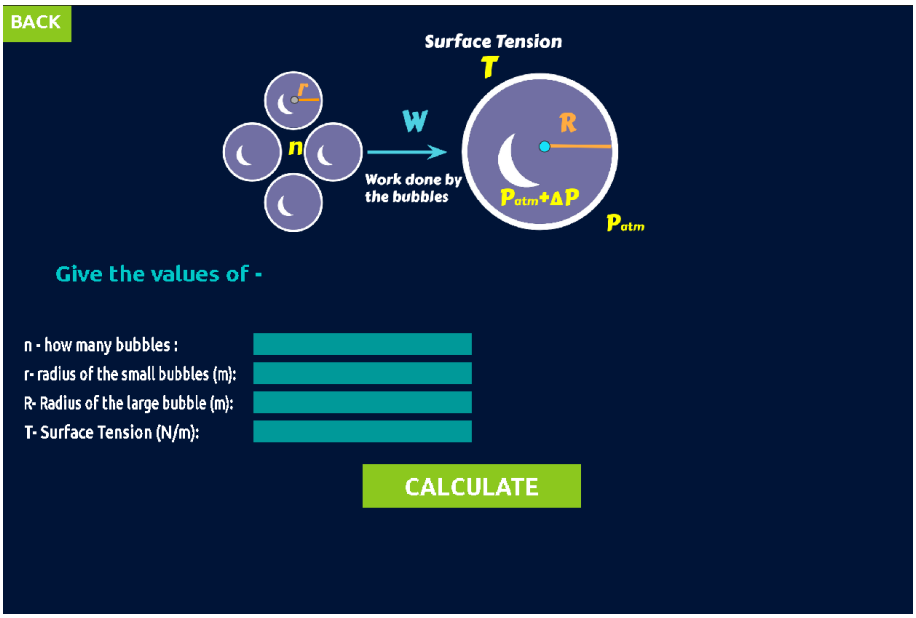






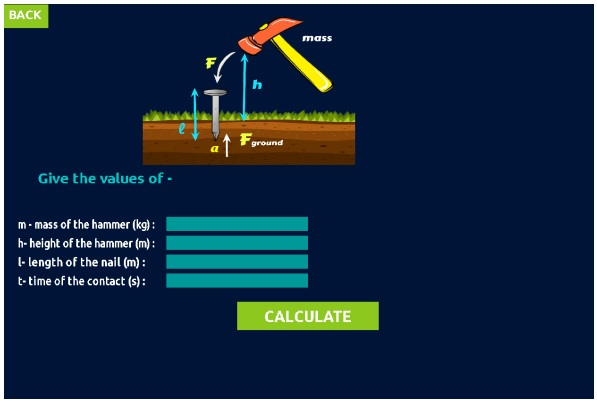


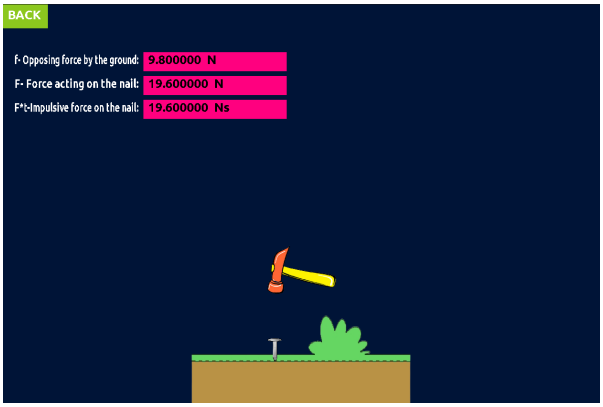
- **8. Surface Tension**
  The problem page-



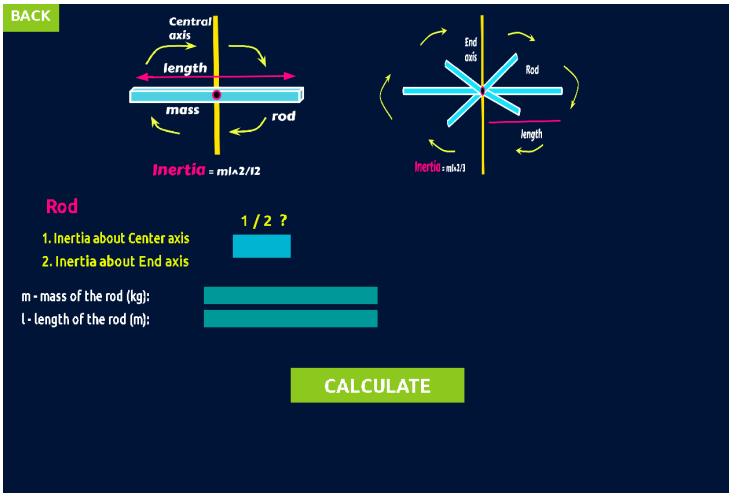
The calculation, animation page-

- **9. Hammer & Nail**
  The problem page & the calculation, animation page-

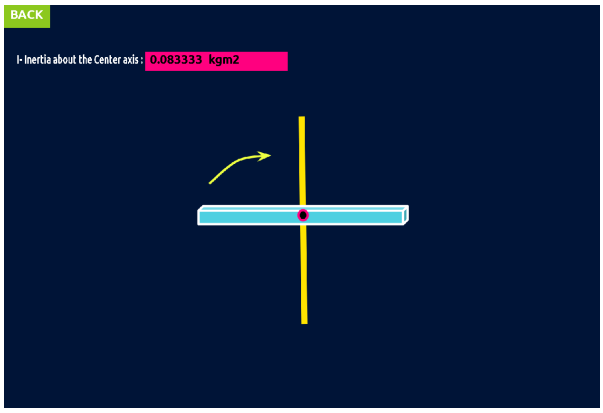


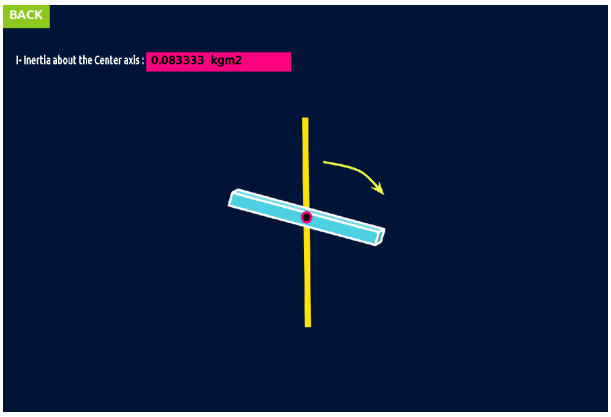


- **10. Inertia of Rod**
  The problem page- here are 2 options. The center and the end axis.

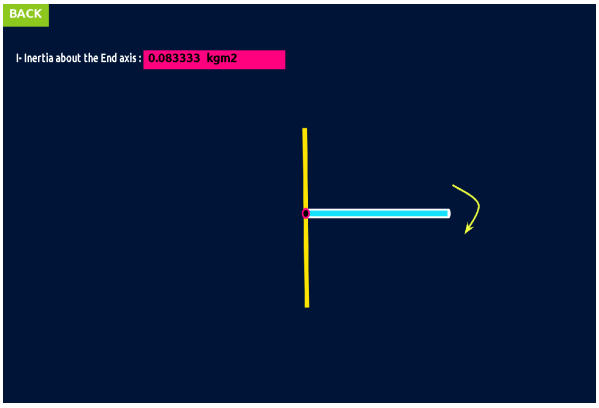


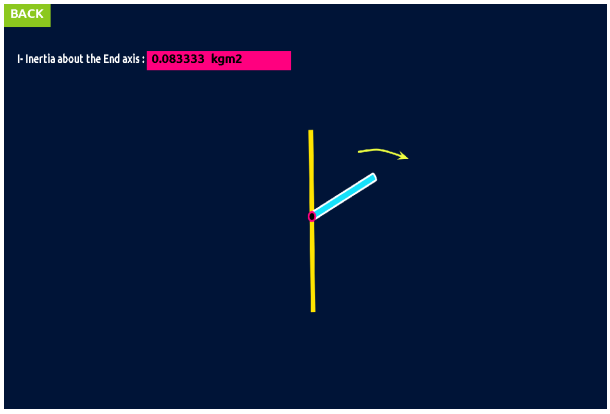
The calculation & the animation is different for the options.
Option 1 animation page-



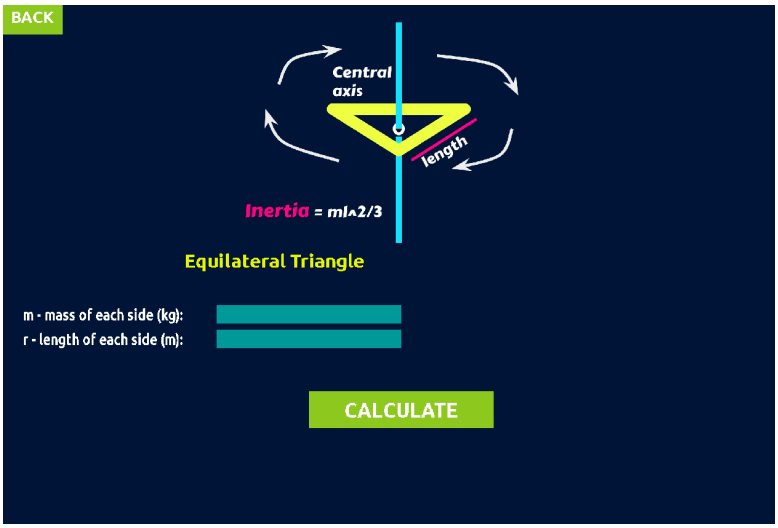


Option 2 animation page-

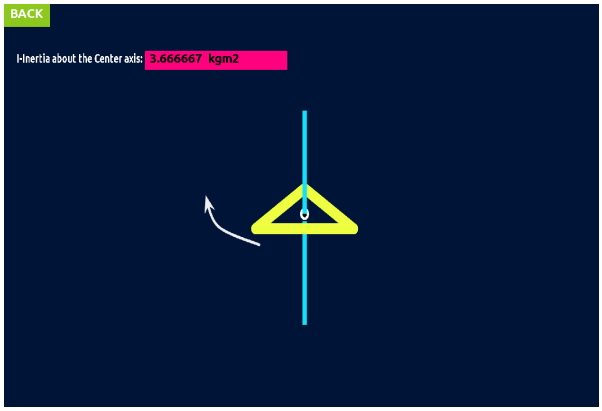


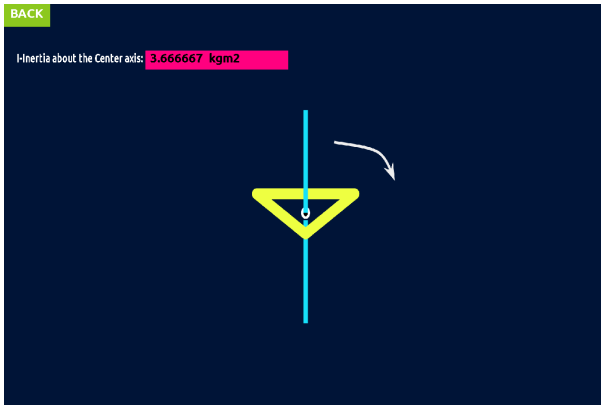


- **11. Inertia of Triangle**
  The problem page-

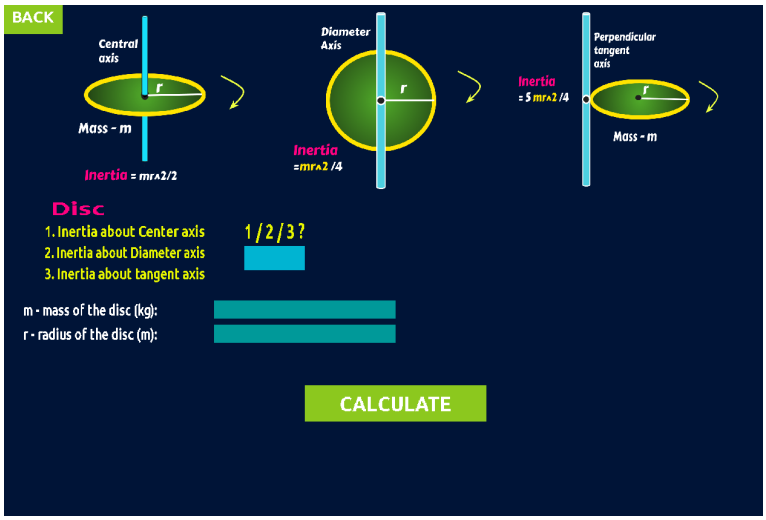


The calculation & animation page-

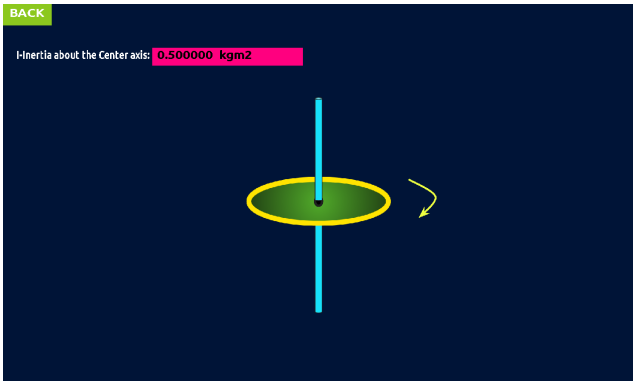




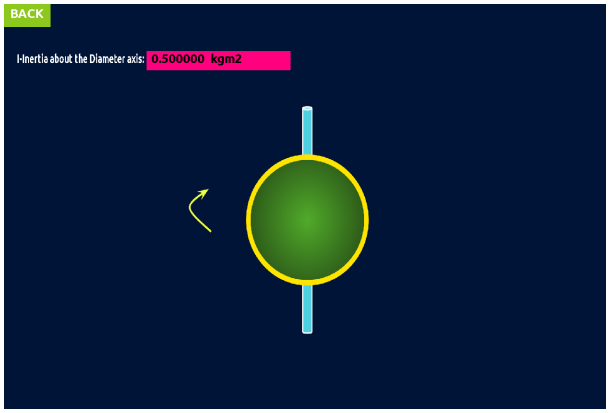
- **12. Inertia of Disc**
  The problem page-here are 3 options. The center, the diameter & the tangent axis.

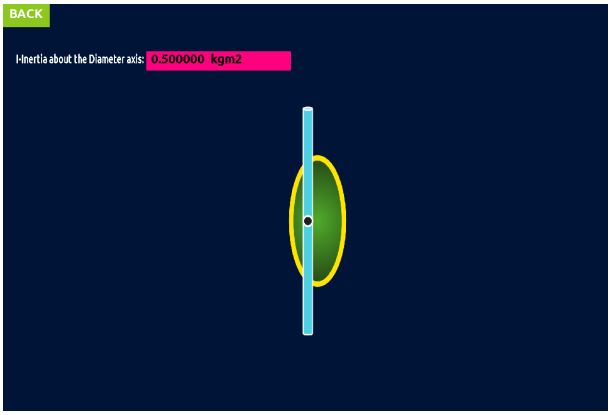


The calculation & the animation is different for the options.
Option 1 animation page-

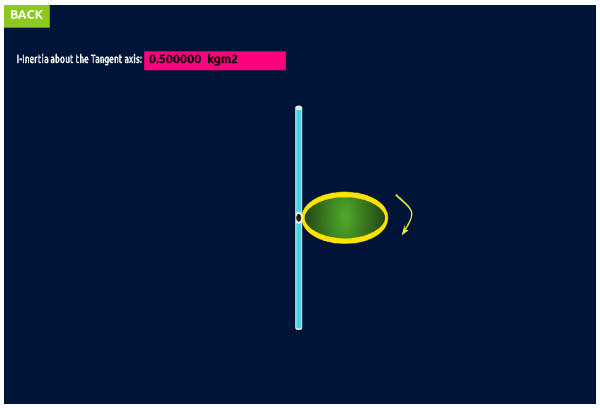


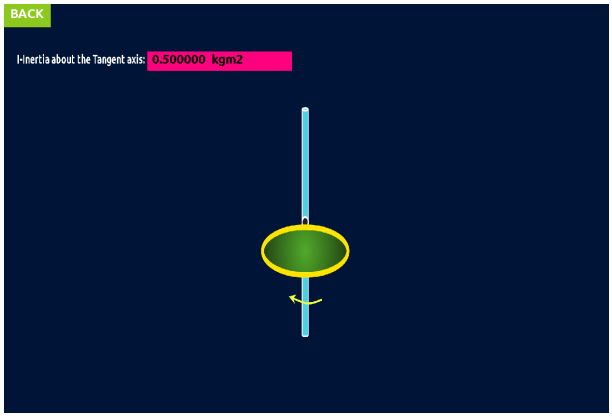
Option 2 animation page-



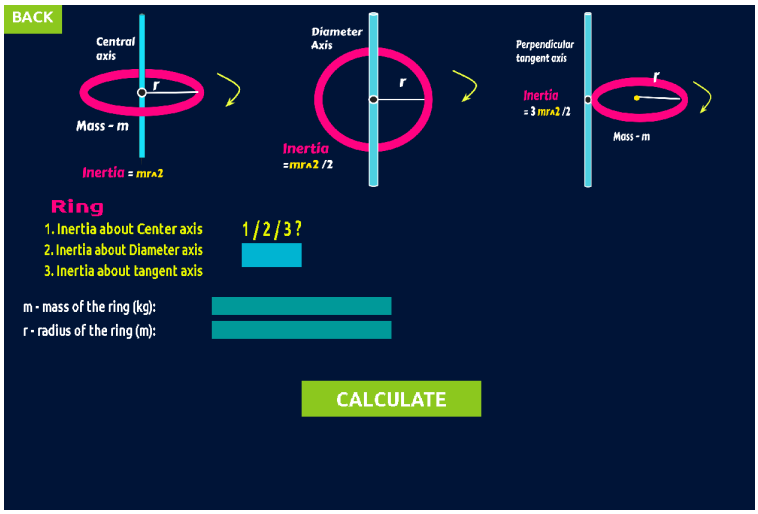


Option 3 animation page-



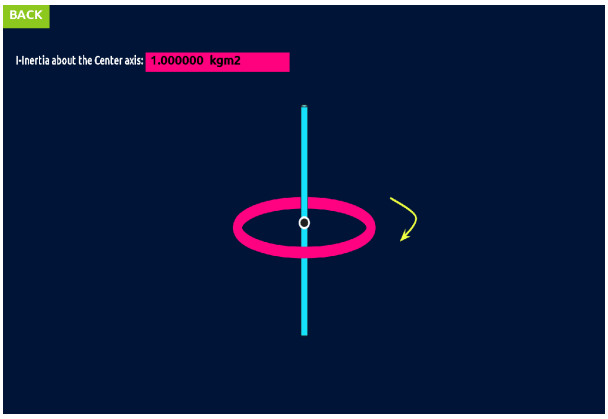


- **13. Inertia of Ring**
  The problem page-here are 3 options. The center, the diameter & the tangent axis.

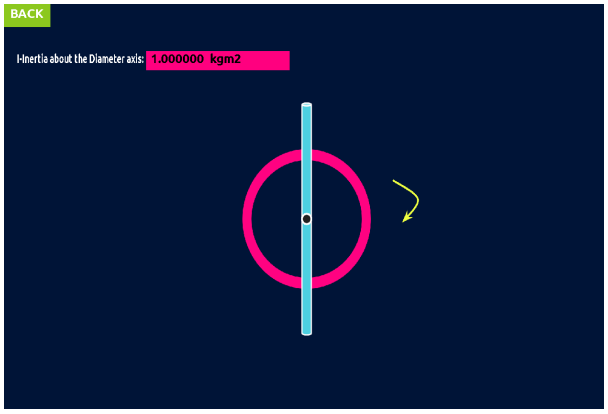


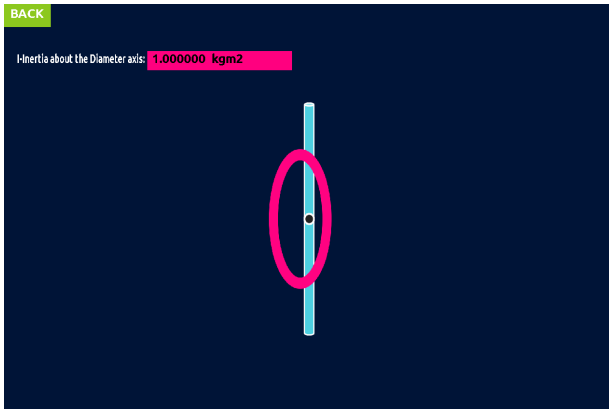
The calculation & the animation is different for the options.

Option 1 animation page-

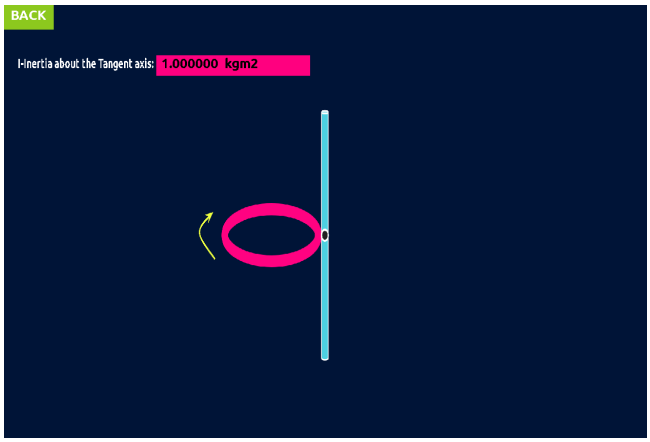


Option 2 animation page-

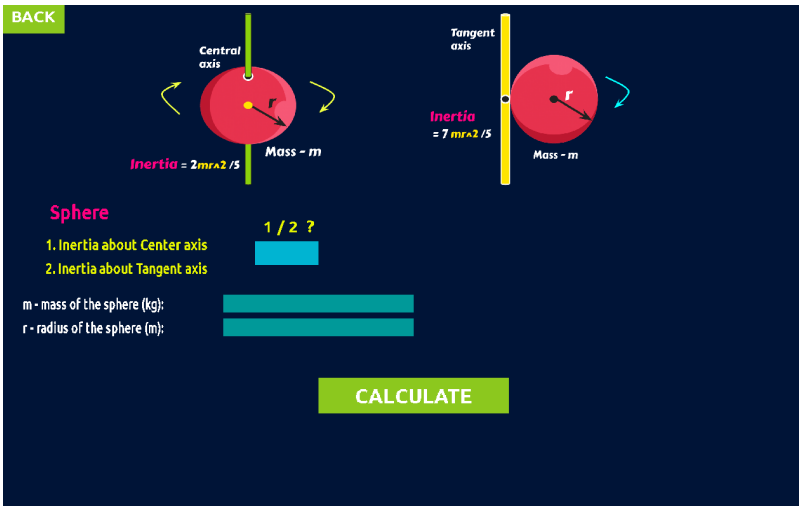




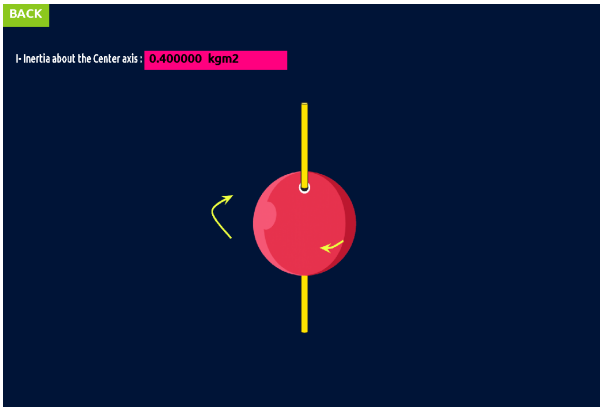
Option 2 animation page-



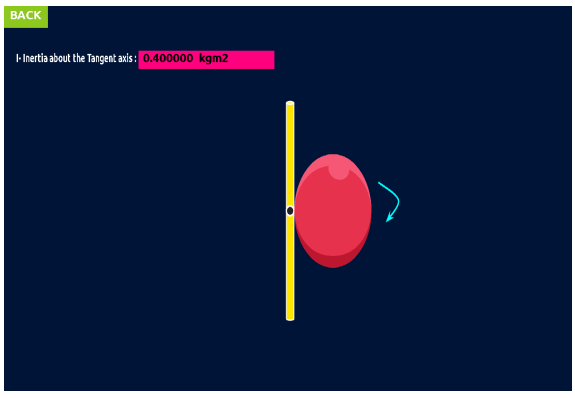
- **14. Inertia of Sphere**
  The problem page-here are 2 options. The center, the diameter axis.

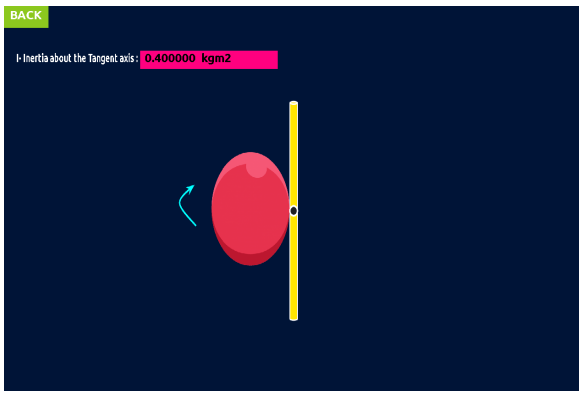


The calculation & the animation is different for the options.
Option 1 animation page-



Option 2 animation page-





Project Modules

1.main.cpp : Calling the main function with the SDL function init(), which initializes window creation, starts ttf and initializes the image . Then calling update() function, mainly defined in header file update.h, which includes image rendering and other initializations with the updating sector of our project. Finally calling the finish() function we terminated window creation, destroyed texture and other SDL stuff.

2.image_ren.h: Here we called the Load_image function which basically loaded image texture.We set a SDL_Rect array which stored different images.Also we did some button special effects here.

3.update.h: This is our main update file which was used as kind of mother files.We defined functions like init(),finish() and also it is used as the main loop of the project.We created mouse event for calling the next function defined on certain header files and included them as user defined header files.

4.option.h: This is for the option part of our different problem sets related to mechanical calculator.Here we added option sections for our inclined plane, impulse and atwood machine problem. We defined different event functions for interacting with users to navigate through the problem.We created buttons and kept mouse events for using those.

5.problemINP.h: Here we defined all the functions for our "Inclined plane" problem sections. We defined functions for different input for the problem, created buttons to interact with the users.There are functions for various branches problems related to inclined planes.

6.CalculateINP.h: This is the calculation section for the "inclined plane" problem. We defined functions for calculation and animation for the problem.The animation section contains a triangular shape with a

box that moves on the side of the triangle due to gravity. And the shape changes according to the value of angle.

7.ProblemImpulse.h: There are functions for our “Impulse(very popular mechanical problem)” problem section.Users can give inputs according to their needs to solve very difficult problems in a second.

8.CalculateImpulse.h: Calculation section with animation for the “Impulse” section. For animation we have a wall, with a ball which collides with the wall at a certain angle given by the user and bounce back at an angle according to the laws of physics.

9.ProblemAtwood.h: Here we defined all the functions for our “Atwood Machine” problem sections.Atwood machine is a very popular problem related to pulley systems.This section is used for the user inputs.

10.CalculateAtwood.h: Calculation section with animation for the “Atwood Machine” problem section.For animation we have two boxes hanging with the roops due to gravity.And the boxes move along the pulley for gravitational force .There works tension force due to the tension of the roops.

11.problem_Rotational_Moton.h: From here we started our rotational mechanics. We defined functions for our different input for the rotational motion.

12.calculate_Rotational_Motion.h: For the animation part we rendered a ball and rotated it around a circle and the radius of the circle changes according to the input given by the users.There are also calculation portion for the problem.

13.problem_Angular_Momentum.h: We defined functions for users to give different inputs for calculating the angular momentum.

14.calculate_Angular_Momentum.h: Calculation and animation section for the “Angular Momentum” problem. In the animation a ball changes its speed and circulating radius according to different inputs.

15.problem_Hammer_Nail.h: Back to old school work energy physics problem.We created functions for very popular conservation of energy and work-energy theorem example problem hammer nail.This section is for user interaction.

16.calculate_Hammer_Nail.h: This is the calculation and animation part of the problem section of “Hammer Nail” problem. For animation we have a hammer(!) and a nail that goes deep through the ground according to the falling speed of the hammer(which changes with the user input).

17.problem_Surface_Tension.h: Newtonian mechanics is imperfect without surface tension.So here we go with our surface tension problem.We did it for bubbles of different density,where we calculated work done for surface tension to join more than one bubbles to make one!

18.calculate_Surface_Tension.h: This is the calculation with the animation portion for the problem part.For calculation we rendered an image of a bubble and showed bubbles adding up to make a big bubble.The radius and the number of bubbles change according to the user input.

19.problem_Depth_Measurement.h:A bubble of water goes up to the surface of water from the bottom and its volume increases.From the increasing volume and radius we can measure the depth of the water!We implemented this interesting problem here.

20.calculate_Depth_Measurement.h: Here is the calculation and animation part for the problem.We calculated the depth and animated a

bubble increasing by its volume to reach at the surface of the water from the bottom.

21.problem_Wave_Melody.h: Now here it comes- waves! We calculated the frequency ,wave number,length,and velocity of the wave of a melody.

22.calculate_Wave_Melody.h: We calculated answers for different values given by the users and animated waves for different frequencies.

23.problem_Inertia_1_rod.h: Here we implemented the functions for the inertia problem related to rod.The users can give input for the rotational axis also from which part they can rotate a rod.

24.calculate_Inertia_1_rod.h: This is the calculation and the animation section for the problem.For the animation we rendered a rod image and rotated it for different rotation axes.

25.probem_Inertia_2_Triangle.h: This is exactly the same version for the rod problem except it does it for a triangle.We implemented necessary functions for that.

26.calculate_Inertia_2_Triangle.h: This is the animation and calculation part for the related problem.We rendered an image of an equilateral triangle and rotated it through the different axes given by the users.

27.problem_Inertia_3_Disc.h: There are functions for the user input for the problem related to inertia of a disc for different axes of rotation.

28.calculate_Inertia_3_Disc.h: This is the animation and calculation part for the Disc problem.For animation we rendered an image of a thick disc and rotated it for different axes given by the users.

29.problem_Inertia_4_ring.h: This is exactly the same version for the disc problem except it does it for a ring.And the ring is hollow.We implemented necessary functions for that.

30.calculate_Inertia_4_ring.h: We did this for calculation and animation part.For animation we rendered an image of a ring and rotated it for different axes given by the users.

31.problem_Inertia_5_sphere.h: Final problem portion for our project till now, where we implemented the functions related to calculate inertia of a sphere for the different rotated axes.

32.calculate_Inertia_5_sphere.h: And this is our last header file till now where we implemented the calculation portion and the animation part for the inertia of the sphere.For the animation we rendered an image of a sphere and rotated it for different axes given by the users.

Team Member's Responsibilities

1. **Md. Ehsan Uddoula Pahlowan** (Team Leader)
    - Logic design and implementation.
    - Code testing and bug fixing.
    - Menu bar design and coding
    - Structured source code design in C and C++.
    - Version control(Git/Github)

2. **Kanij Fatema** (Team Member)
    - Graphics & background design- (photoshop, image design)
    - Logic design and implementation
    - Font style, menu buttons & appearance design
    - Structured source code design.

3. **Md Shariful Islam Rayhan** (Team Member)
    - Structured source code writing in C and C++.
    - Code testing.
    - Logic design.
    - Implementation and function design.

Platform, Tools & Libraries

- **C and C++** : Basic coding of the project is done in both C and C++.
- **Codeblocks :** Free, useful & powerful IDE.
- **SDL2 :** cross-platform development library designed to provide low level access.
- **Git/Github :** software development and version control using Git.

Limitations

- We have to work more on animation to make problem solving more interactive.
- Written in C and C++ in a structured manner, no object oriented feature is used.
- We have to do more work on file systems to store data.

Conclusion

Apart from learning new language and technology,this project taught us collaboration, pressure handling, peer communication and many other important qualities a software engineer should have.We firstly faced some problems like taking text input, which took a few times to figure out.Then we had problem to do some

animations.But we learned to overcome those difficulties and made our project good overall. We hope we could implement these learning in our life in future.

Future Plan

We will dynamic our project more. We have a plan to show the detailed solutions and set quiz options also. We will do the animation part better. Also we will add more mechanical topics to help the students. The project till now is only on the basics of structured programming. In our future plan we will introduce the application of Object Oriented Programming.

Repositories

Github Repository:
https://github.com/EhsanUddoula/Mechanical-Calculator

Youtube Video:
https://youtu.be/FcnR_1tIyXo

References

SDL Learning Sources:

- *https://www.libsdl.org*
- *Welcome - parallelrealities.co.uk*
- *https://lazyfoo.net/tutorials/SDL/*
- Youtube videos.
- Font Sources:*Ubuntu - Google Fonts*