**Project Report CUBENTS**

**Comprehensive Utility Board for EquatioN Text and Shapes**



**Under the Guidance of**

**Prof. D. B. Phatak**

**Mr. Avinash Awate**

**Mr. Rajanikant Jangir**

**Department of Computer Science and Engineering**

**Indian Institute of Technology, Bombay**

**Certificate of Approval**

Summer Internship 2017 Project Approval Certificate

It is to certify that the following students successfully worked on and completed the project “CUBENTS” under our guidance and supervision

* Bhagyashree Rawal
* Ruchita Parmar
* Surbhi Inani
* Abhijit Roy
* Akash Singh
* Arun Nainwal
* Ankur Singh

Rajnikant Jangir

**Mentor**

Mr. Avinash Awate Professor D. B. Phatak

**Supervisor Guide**

**ACKNOWLEDGEMENTS**

It is a matter of great pleasure for us to acknowledge our deepest sense of gratitude to our mentor Mr. Avinash Awate for his unparalleled vision and ideas. His guidance and valuable suggestions helped us in the completion of the project. We also have the privilege of thanking Prof. Dr D.B. Pathak, CSE Department, IIT Bombay for selecting us to be a part of this internship programme at this esteemed institute which provided us with the various facilities during the course of our project. We also thank him for inspiring us through his tireless efforts in spreading quality engineering education. Thanks are due to Rajanikant Jangir, Sr. Software Engineer, for his timely help in all technical issues we encountered throughout the course of the internship. We also thank the IITB Administration for making our stay here at IITB memorable and pleasant.

**OUR TEAM**

* Bhagyashree Rawal (Sarvajanik College of Engin. and Tech.)
* Ruchita Parmar (Mumbai University)
* Surbhi Inani (Carnegie Mellon University)
* Abhijit Roy (IIT Patna)
* Akash Singh (NIT Warangal)
* Arun Nainwal (NIT Silchar)
* Ankur Singh

**CONTENTS**

**INTRODUCTION**

We believe that the government alone cannot meet the needs of 400 million children under the age of 18 and we need a concerted public initiative to provide a glimmer of hope to millions of underprivileged children.

**Current State of Online Learning**

Massive Open Online Courses (MOOCs) is overpowering the area of online education everywhere. Professors from world class institutes are making online content (and assessments) available both in free and (increasingly) paid courses. Renowned institutions are accepting these credits as part of their curriculum. However, most of these courses are for under-graduate level subjects. The situation in India is slightly different. A survey suggests that the basic education system is at fault. Our opportunity and our obligation is to reimagine our schools and give all children an education that will help them thrive in a world with a strong foundation of basics.

We believe that the present situation is only a transitory phenomenon. Learning is all set for a revolutionary change. We believe real student centric learning will finally arrive in a decade or so. Any individual (including a child) will be able to choose what he/she wishes to learn, what degree of mastery he/she wishes to achieve in the chosen field, when he/she wishes to acquire the knowledge and at what rate he/she is comfortable in acquiring the knowledge. There will be no certificate examinations like our X and XII boards, instead each 'subject' will have its own definitions of mastery and its own certificates. Such a freedom to learn is now available in 'hobby' fields such as dancing, singing, cooking etc. One of the key ingredients to facilitate such learning is absorption of technology in the process of learning. The current MOOCs attempt to bring the classroom to the learner's laptop. The next generation content will expand the learning to what is not possible in a classroom. The other key ingredient is the use of technology to assess a learner's performance by automatic evaluation of learner assignments. By providing automatic assessment reports on completed tasks, the learner gets immediate feedback and can then resubmit the task. The aim of this project, too, is to bring the world one step closer to this goal.

We aim at improving the online learning content for school going students.

**Target group**

Presently courses in MOOCs generally target undergraduates and above levels.There are hardly any assessments for school going students for grades 7th-12th. Our tools are aimed at assessing such students of grades 7th to 12th when the fundamentals of the student can be developed and strengthened. Our approach: interactive learning and assessment Our tools fully involve learner and hence ensure maximum understanding of the concept. The learner is allowed to change parameters of the physical set-up and see the effect immediately. This allows the learner to understand certain phenomena which cannot be demonstrated even in school laboratories. The learner can verify the concepts learned through textbooks or classroom.The current state of MOOC is such that it teaches the learner through the medium of videos.

With CUBENTS we aim to bring static learning content to life. Thus learning becomes fun for and retention is increased.

As Benjamin Franklin rightly said- “Tell me and I forget, teach me and I may remember, involve me and I learn”

**PURPOSE**

The purpose of this document is to provide the project report for the EkShiksha Project for the Ekalavya Summer Internship Program 2017 conducted at the Indian Institute of Technology, Bombay.

**Objectives**

* To provide a ready to deploy product to directly benefit the students and teachers in a measurable manner.
* The project implements a smart white board where educational content can be displayed.
* The primary objective is to provide an online learning platform for students.
* The other objective is to provide a simple user-interface for the teachers to create lessons.
* Using this platform students will learn fundamental concepts of any subject in a visual and interactive manner.

**Project Scope**

The purpose of this project is to create a convenient and easy-to-use interactive web application for teachers and students to primarily teach and learn algebra, 2D and 3D geometry and statistical mathematics. Other subjects can also be taught. The program is based on a smart web whiteboard template utilizing and incorporating the relevant features of paint, Microsoft Word and Powerpoint. It also creates a generic platform to write or upload equations which can be given animation effects and can later be viewed on the student interface. Above all, we hope to provide a comfortable user experience along with maximum learning and retaining of concepts.

**Intended Audience**

This project is useful to all teaching institutions and professors teaching subject courses and also to students between class 1 and 12 who want to learn and practice concepts in an interactive and animated setting.

Any inquisitive person can also contribute by creating educational content and saving it for others to view and learn from.

**Importance of e-Learning**

**Standardisation**

With e-learning everyone gets the same content, presented in the same way. All users regardless of time and place in which they are trained, have access to quality content. This helps to prevent the loss of information from one course to the other, guaranteeing that training’s consistent.

**Easy Learning content can be updated rapidly**

E-learning provides organization with a greater ability and capacity to manage major changes more quickly. A course can be easily modified and it can also adapt itself to changes in content in a short amount of time.With e-learning the user has the freedom to work on his own pace.He may go slower or faster based on his own understanding of the topic rather than the traditional way of student-teacher learning where the student has to adjust to the teacher’s pace of teaching and thus may not fully grasp the topic.With e-learning the student does not need to worry about his own grasping power and this is a major advantage of e-learning.

**Interactivity**

The user of an eLearning course is not a passive human being that likes to receive information, it is someone that likes to practice, analyze data and perform tasks. If they are provided with a virtual platform or Learning Management System, he or she will be a part of online forums, chats wikis and workshops. There is a better understanding of what is being taught when a person reads the text, listens to the information, looks at the simulations that will back it up and practices what has been learned.

**Anywhere and Anytime**

Having eLearning based training is ideal for the new collaborators that need high quality, immediate knowledge. It gets rid of the hassles of having to plan an on-site course or having to wait around for more people to join the course.Even in remote areas where quality education is not available,e-learning brings the school at the doorstep of the student.

**Economic**

eLearning is definitely economically viable as internet is getting cheaper and widely available.Most of the courses provided are free of cost and provide quality course content and assignments. With easily available courses and free education the country’s literacy is growing exponentially.As a result it also improves the economic condition of the country which is a deciding factor in the development of a country.

**Future Scope**

eLearning is a growing entity and in the coming years we may even see almost everything turning towards e-education.From engineering to cooking to almost any field you will find a major hand of e-education and learning in the coming years.

As the famous writer Donna J.Abernathy quoted- "Online learning is not the next big thing, it is the now big thing.”

**IMPLEMENTATION**

The project consisted of the following building blocks:

**Front-end:**

Editor and Learner View

Equations

Graphics

Text

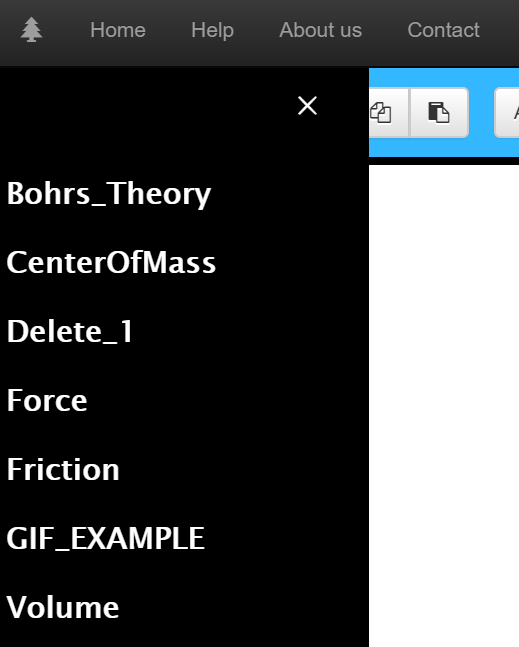
**Back-end:**

Database

**EDITOR AND LEARNER VIEW**

The UI Design is one of the key foundations of the entire project. It includes the entire architectural basics required to have text and canvas graphics co-exist with one another. Some of the salient features that the UI brings to the project apart from look and feel -

1. **Project Tree** - The project tree denoted by the tree shaped symbol on the top-left corner of the screen allows users to load/view any of the saved activity . It provides flexibility in terms of switching in between different tutorials/topics and also allows the user to get an overview of which activities are stored.



1. **Layered Architecture -** The reason for the existence of graphics on canvas and text in HTML div’s is this architecture. This architecture contains a parent div which encapsulates a number of layers. For the sake of flexibility we have kept the canvas on the bottom and then superimposed the other layers of text and equation on top of it. All the canvas graphics are directed to the canvas layer and text/equations are directed to their own layers.This makes it possible to have graphic elements and text elements mingle with each another.

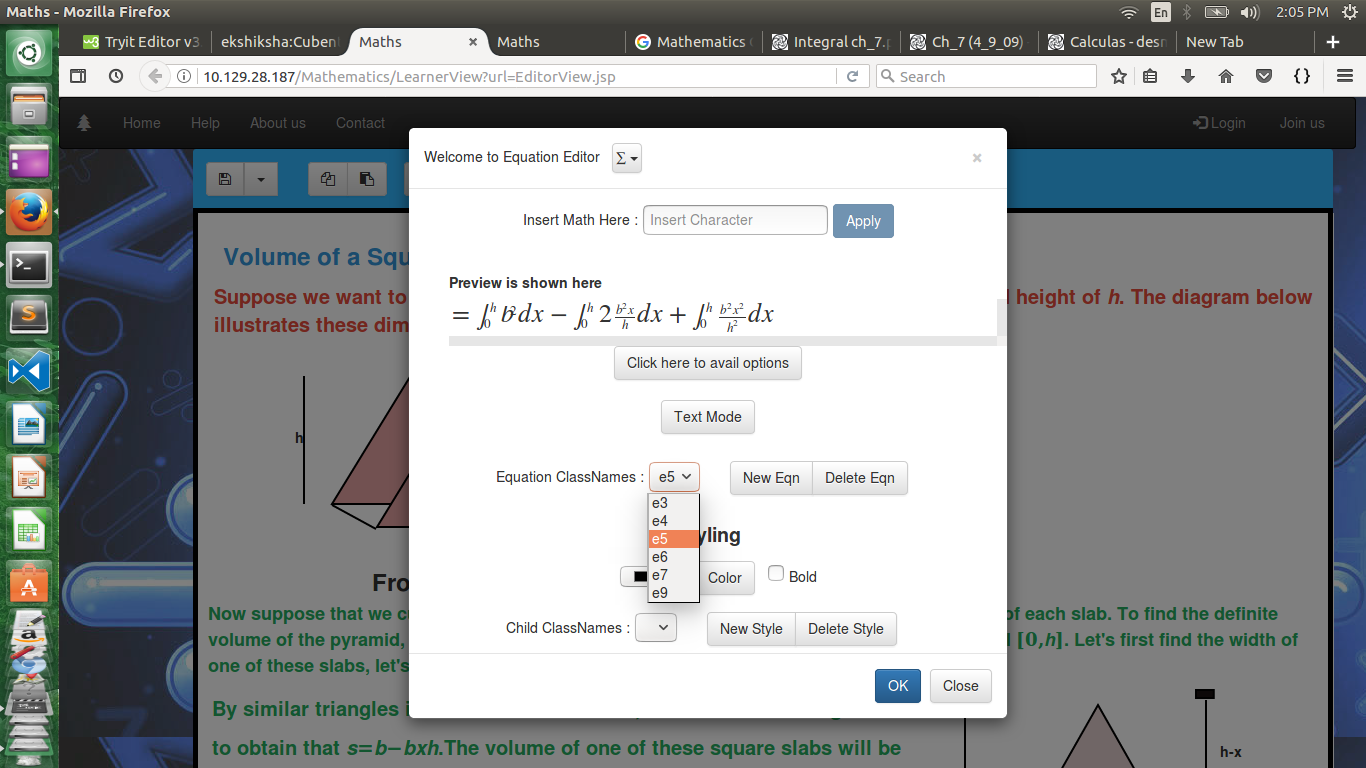
**EQUATIONS**

One of the main features of the entire project is the in **built equation editor** which lets the user create equations using MathML which is a markup language gaining popularity and also becoming the standard markup for web based equations.

Most of the equations which you see online are in fact not native HTML. They are images or they are embedded in a big image. Considering that there a certain user is required to display more than twenty equations on a page, then the condition of the UI developer/user becomes troublesome because he/she must then have to create the images of 20 equations using some external tool and then link the images to the HTML page. This comes with an added disadvantage because images cannot be styled or modified once created let alone animate the equations in the image. Our interface can be seen as a solution to this traditional and now outdated method of rendering equations in browsers.

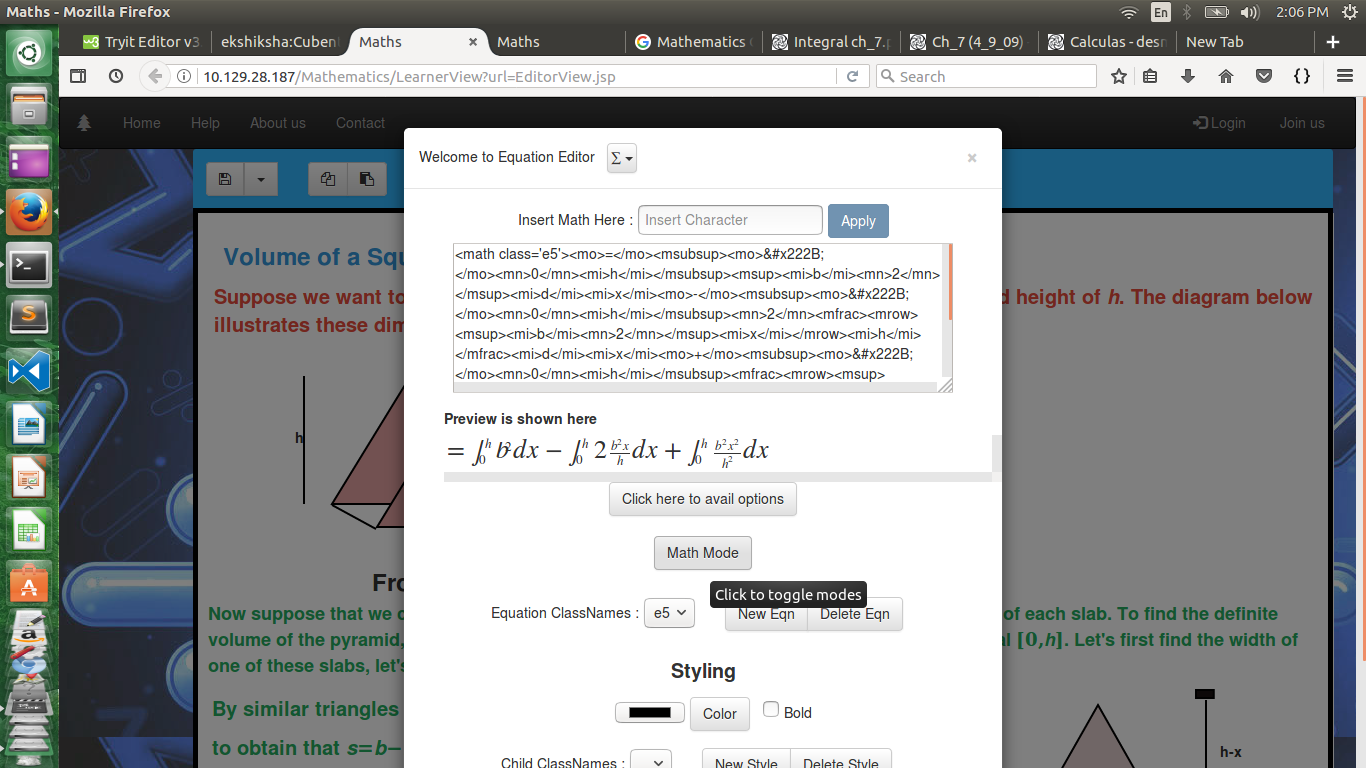
The features of the editor include the following -

1. It allows the user to type in all the variables and operators in a small text-box which is then processed to output the MathML code onto the editor and render the XML on the screen from the MathML generated on the editor.

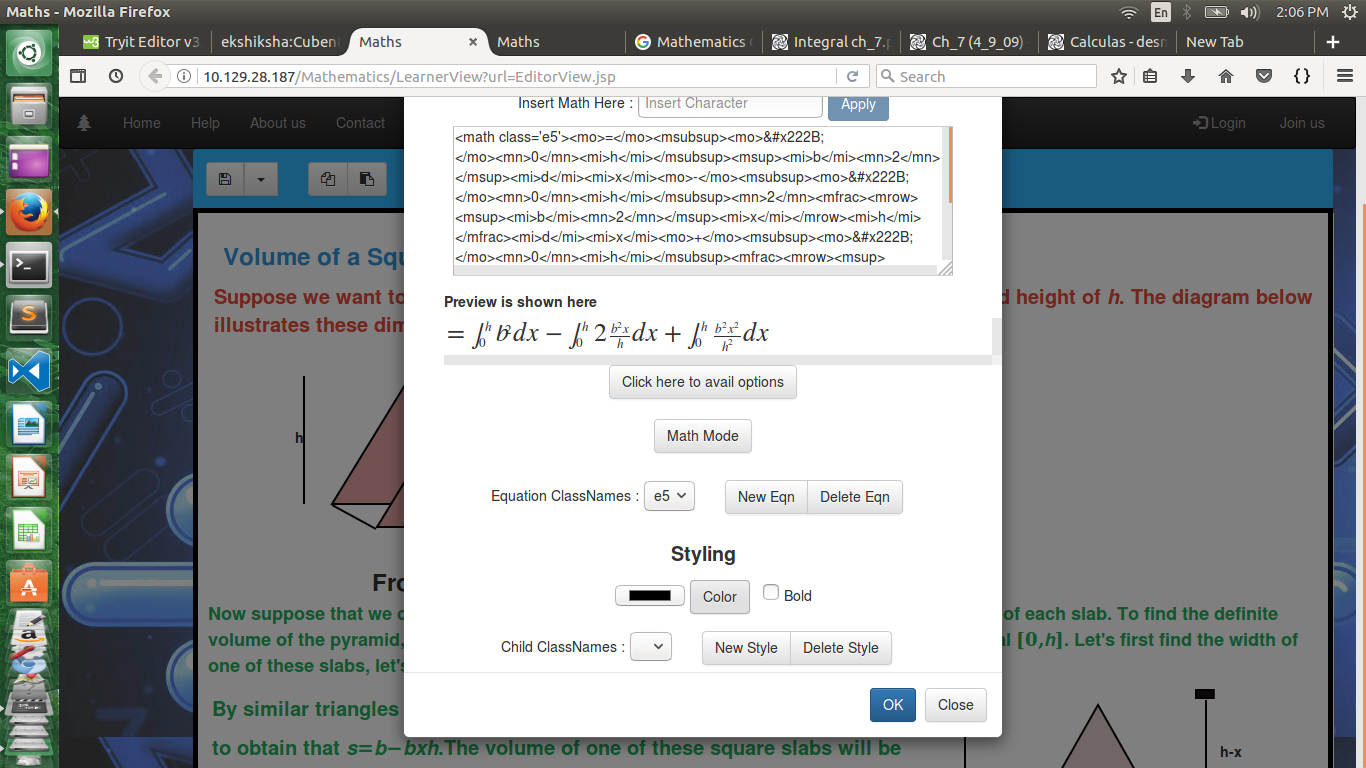


1. It is almost like an **IDE for MathML equations**. The user can create as many equations as possible, style the equations, edit already created past equations and most of all add animations to the created equations.
2. The equation editor allows the user to **import or copy paste MathML codes** to generate equations using MathJax.

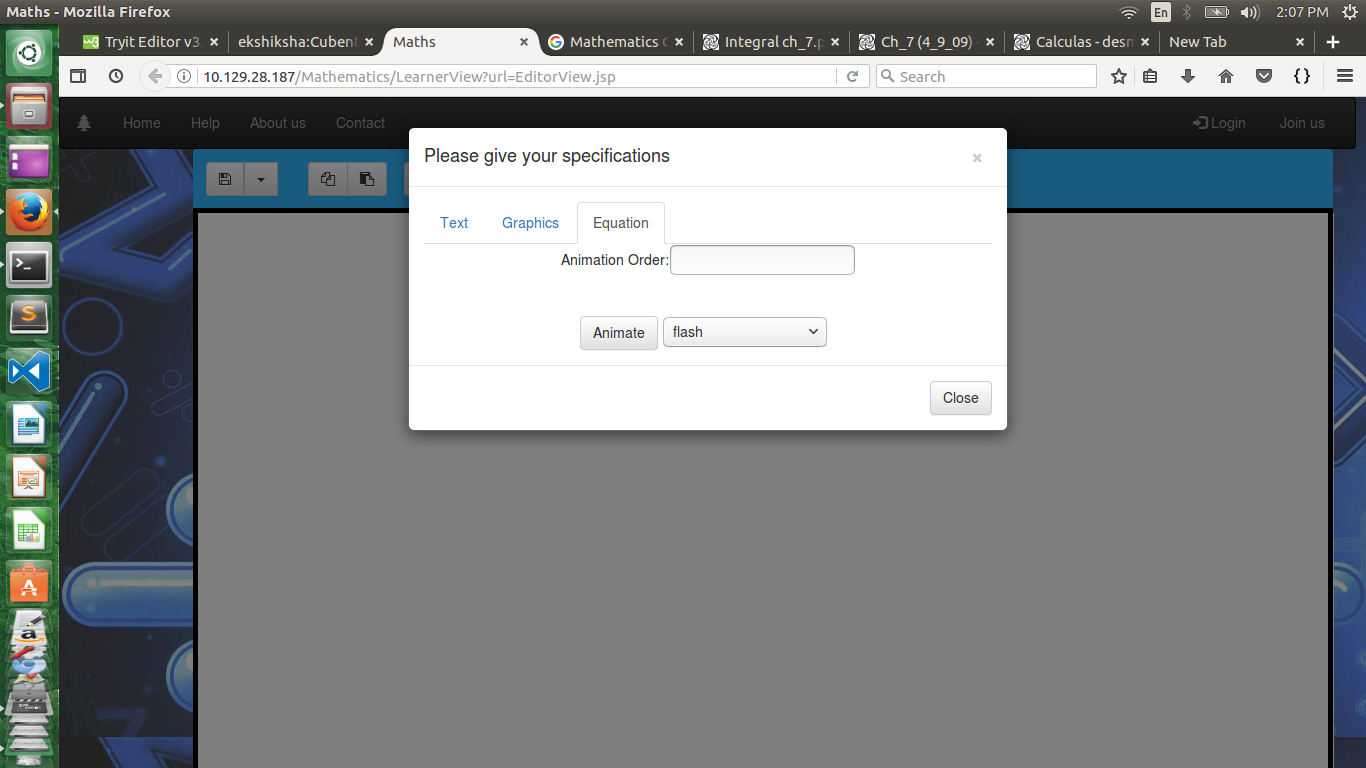
Any code in MathML can be copied and pasted directly on the text-box given and the MathJax rendered code will be generated on the preview. This provides for creating equations as quickly as possible.



1. One of the greatest advantage of using MathJax is the fact that it has a large number of output processors which can output in any format possible except for outputting on the canvas. MathML although is self sufficient in terms of creating equations does not allow other HTML tags to mingle with it’s own tags. Apart from this MathML is not supported by many browsers although it is the web standard. This is where MathJax enters the picture. MathJax is a JavaScript work around to rendering MathML in browsers that do not support it.The MathJax configuration applied in this project is the HTML output processor which processes MathML and converts its various tags into internal library span tags with specific classes for rendering the equation in in HTML on the screen.

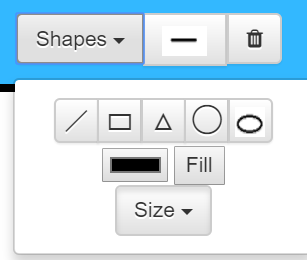


User can also add add new Styles which will incorporate color and bold features. User can also animate the equations.

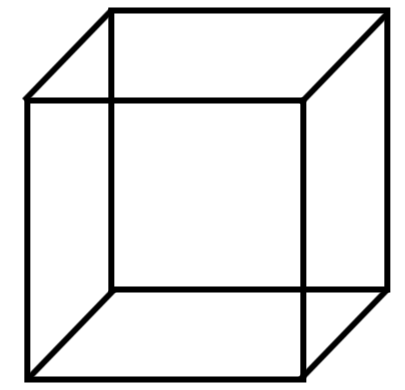
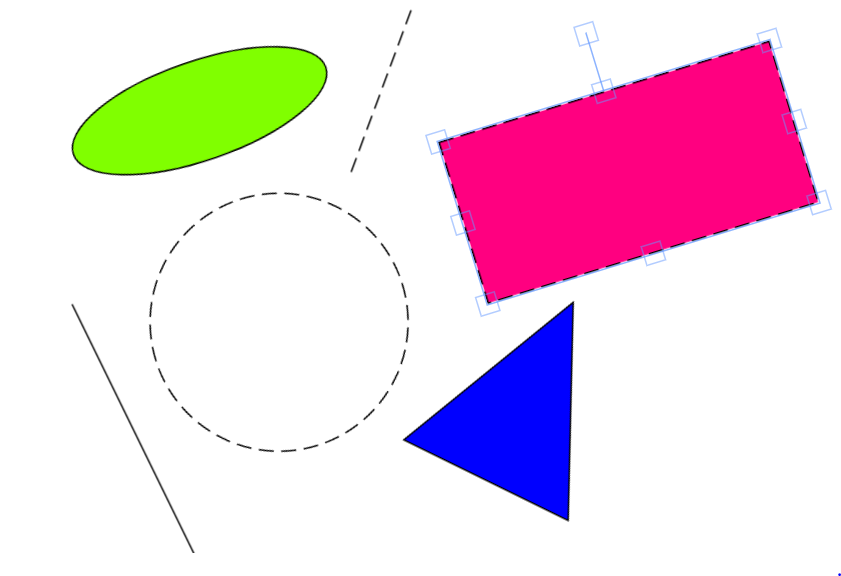


**GRAPHICS**

The Graphics layer of CUBENTS allows the user to draw different shapes on the canvas. 2D shapes like line, rectangle, circle, ellipse and triangle are extremely important in Mathematics and Physics because it covers the topics of 2D and 3D geometry, trigonometry, physical concepts like Newton’s laws, etc. CUBENTS maximizes interactive and visual learning by providing the tools necessary to bring the topics and lectures from our textbooks to life through graphics and animations.

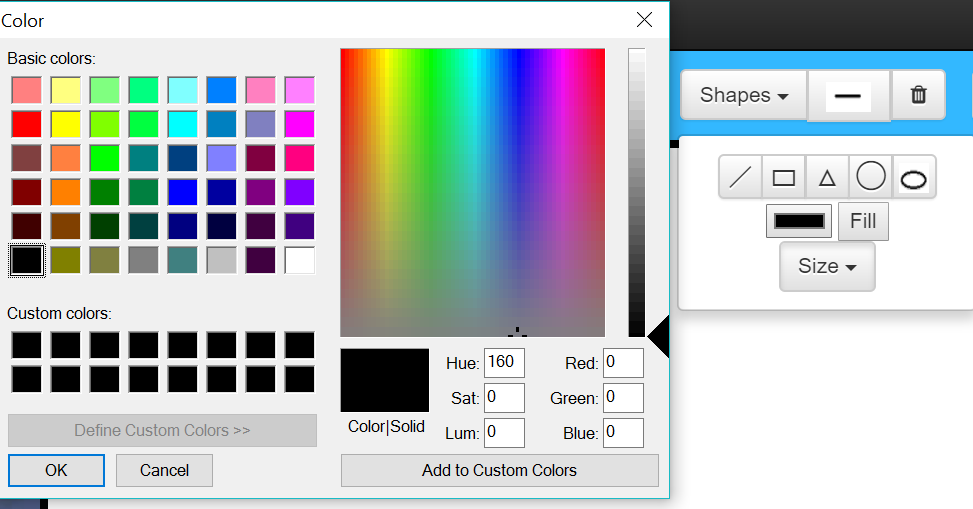


Complicated shapes like cylinders, cones and cubes can be drawn in 2D by grouping various simple shapes together. Shapes like circle, triangle, rectangle, line and ellipse can be drawn by selecting the shape’s button from the toolbar and drawing the shape by dragging the mouse on the canvas to the desired size. The shape element can be removed using the bin button on the main toolbar. Shapes can be easily resized and rotated using the functionalities provided by our graphics package.

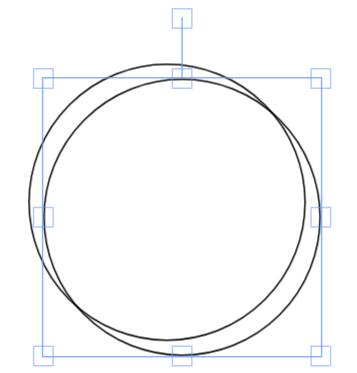


The Stroke type/style of the shapes can be toggled between two options on the main toolbar. Strokes can be continuous or they can consist of a periodic series of dashes and gaps. Each stroke has its own blending mode, which determines how it interacts with other operations in the same group.

 Dashed vs Smooth 



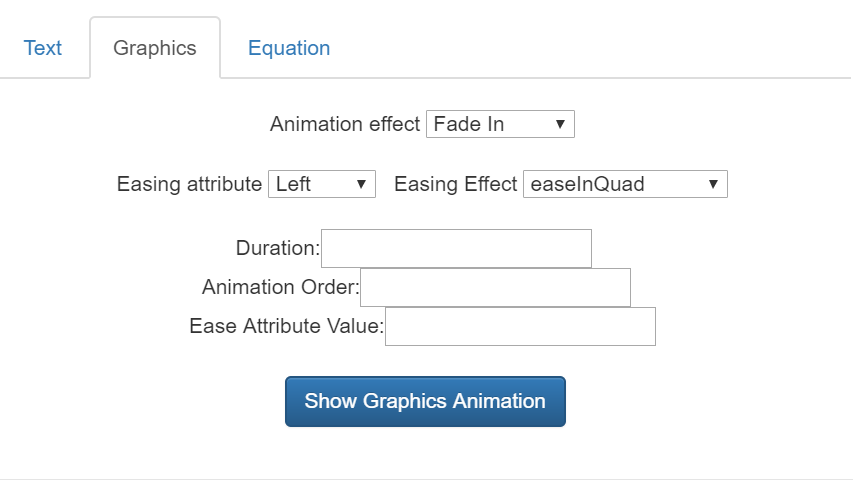
The Fill Operator in the sub menu of Shapes allows the shapes (both smooth and dashed) to filled with any colour desired by the user.



The generic COPY and PASTE buttons work for graphic elements onto the canvas with ease.

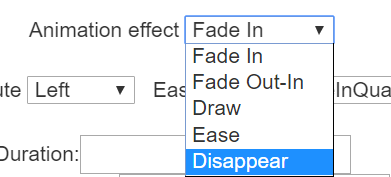


**ANIMATIONS ON GRAPHIC ELEMENTS**



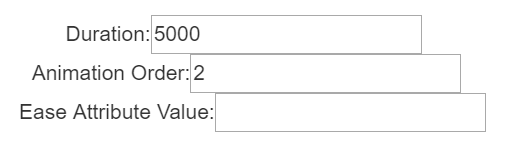
The animations menu for graphics is the second option in the Animations button menu. By selecting a shape element, we can begin to add animation attributes to it.

First, we choose an Animation Effect from the drop down menu.



If choosing the Ease effect, we then specify and easing attribute and easing effect style. Then specify the easing attribute value.

All elements can be given duration and animation order as well that will get synchronized with the timings of the rest of the elements on screen during preview and play time.



Thus, our graphical animations help make our lectures and concepts seem more visual and interesting to learn.

**TEXT**

The third feature that CUBENTS has is a good text editor. With text support user can not only write text but also format it. The Text written can also be given an animation, delay and duration. It can be grouped together with Equation and Graphics as well. Even a group of selected text,selected equations and graphics can be dragged, having text element as a pivot element and the entire group can be dropped anywhere on the board. This feature helps in making better layout of the board easily.

The features of the text editor are listed below:

* User can write text.
* User can format the text using color, size, font family and similar other options.
* User can add table and store it.
* User can insert images from internet.
* User can also insert GIF images.
* The text editor supports WYSIWYG feature.

With strong text support accompanied with graphics and equations, explaining a lesson to a student becomes extremely easy.

The text editor we are using to support all the above features is CKeditor.

1.One can add a text element by clicking on the text button ‘T’.

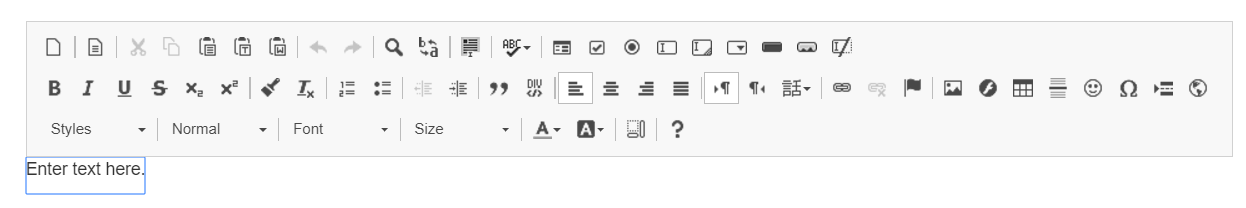


2.One can delete the last selected text element by clicking on the text delete button

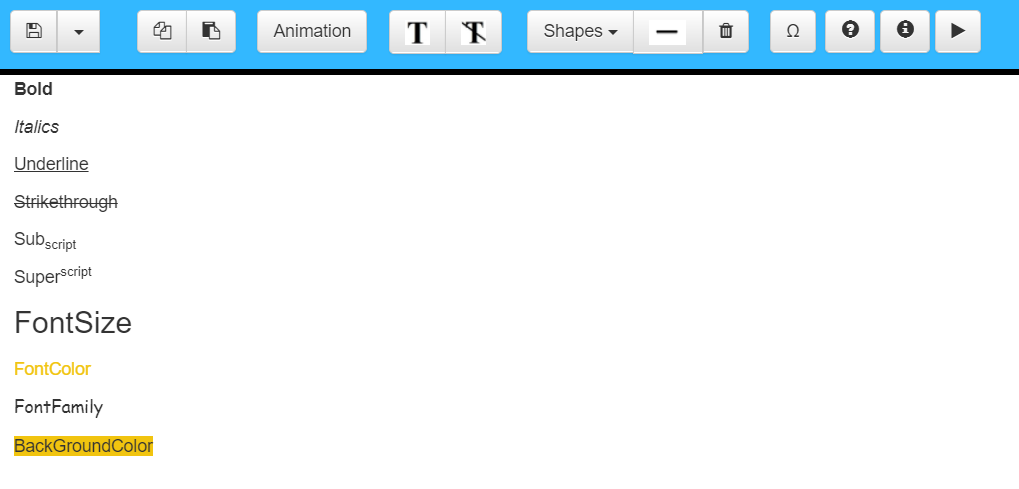


3.Once the textbox is added it can be dragged around on the board.

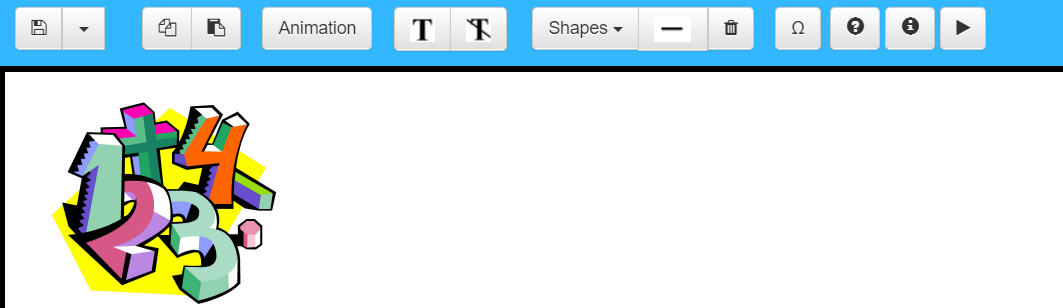
4. By double clicking the textbox it becomes contenteditable and an inline instance of CKeditor opens up as shown in the figure below.

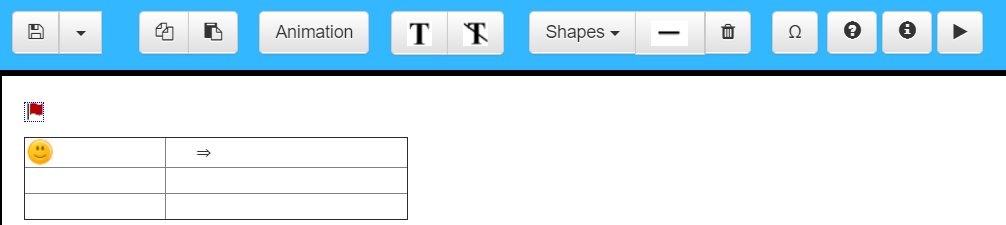


5.With the inline text editor one can get all the options to format text like forecolor,background color, font size, font family, bold, italics, underline, strikethrough, subscript, superscript.



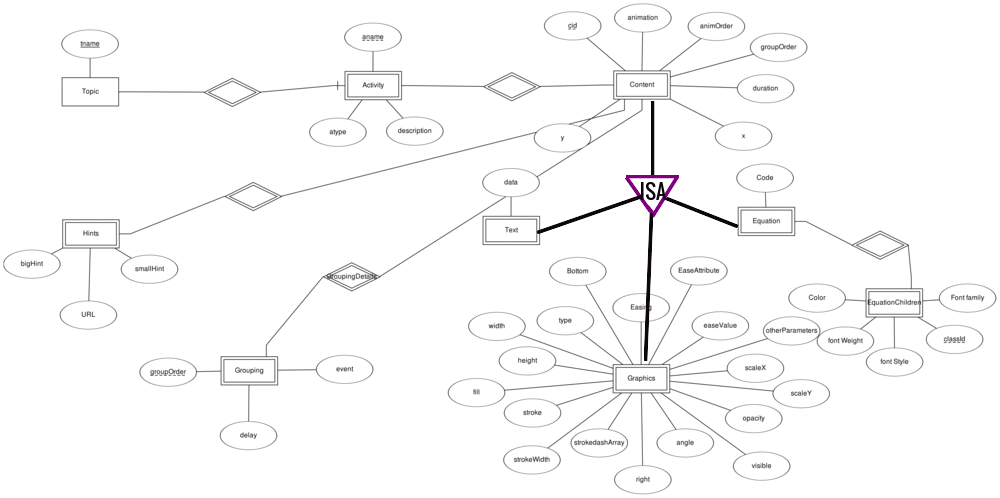
6. With CKeditor one can also add extra images,gifs, tables, flash animations, smileys, special symbols, links.





7.One can also get features like indent and outdent, quote text,searching a word, copy and paste selected text, replace a word with another and so on.

**DATABASE**



The schema can be derived from the queries:

CREATE TABLE `topic` (

`tname` varchar(30) NOT NULL,

PRIMARY KEY (`tname`)

)

CREATE TABLE `activity` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`atype` varchar(30) DEFAULT NULL,

`adesc` text,

PRIMARY KEY (`tname`,`aname`),

CONSTRAINT `activity\_ibfk\_1` FOREIGN KEY (`tname`) REFERENCES `topic` (`tname`) ON DELETE CASCADE

)

CREATE TABLE `content` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`cid` varchar(5) NOT NULL,

`x` double DEFAULT NULL,

`y` double DEFAULT NULL,

`groupOrder` int(11) DEFAULT NULL,

`animation` varchar(30) DEFAULT NULL,

`animOrder` int(11) DEFAULT NULL,

`duration` int(11) DEFAULT NULL,

PRIMARY KEY (`tname`,`aname`,`cid`),

CONSTRAINT `content\_ibfk\_1` FOREIGN KEY (`tname`, `aname`) REFERENCES `activity` (`tname`, `aname`) ON DELETE CASCADE

)

CREATE TABLE `equation` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`cid` varchar(5) NOT NULL,

`code` text,

PRIMARY KEY (`tname`,`aname`,`cid`),

CONSTRAINT `equation\_ibfk\_1` FOREIGN KEY (`tname`, `aname`, `cid`) REFERENCES `content` (`tname`, `aname`, `cid`) ON DELETE CASCADE

)

CREATE TABLE `equationChildren` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`cid` varchar(5) NOT NULL,

`classid` varchar(10) NOT NULL,

`fontstyle` varchar(10) DEFAULT NULL,

`fontweight` varchar(10) DEFAULT NULL,

`color` varchar(10) DEFAULT NULL,

`fontfamily` varchar(30) DEFAULT NULL,

PRIMARY KEY (`tname`,`aname`,`cid`,`classid`),

CONSTRAINT `equationChildren\_ibfk\_1` FOREIGN KEY (`tname`, `aname`, `cid`) REFERENCES `equation` (`tname`, `aname`, `cid`) ON DELETE CASCADE

)

CREATE TABLE `graphics` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`cid` varchar(5) NOT NULL,

`type` varchar(10) DEFAULT NULL,

`width` double DEFAULT NULL,

`height` double DEFAULT NULL,

`fill` varchar(30) DEFAULT NULL,

`stroke` varchar(30) DEFAULT NULL,

`strokeWidth` int(11) DEFAULT NULL,

`strokeDashArray` varchar(30) DEFAULT NULL,

`scaleX` double DEFAULT NULL,

`scaleY` double DEFAULT NULL,

`opacity` int(11) DEFAULT NULL,

`visible` tinyint(1) DEFAULT NULL,

`angle` double DEFAULT NULL,

`riht` double DEFAULT NULL,

`bottom` double DEFAULT NULL,

`easing` varchar(30) DEFAULT NULL,

`easeAttr` varchar(30) DEFAULT NULL,

`easeValue` int(11) DEFAULT NULL,

`otherParameters` text,

PRIMARY KEY (`tname`,`aname`,`cid`),

CONSTRAINT `graphics\_ibfk\_1` FOREIGN KEY (`tname`, `aname`, `cid`) REFERENCES `content` (`tname`, `aname`, `cid`) ON DELETE CASCADE

)

CREATE TABLE `grouping` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`groupOrder` int(11) NOT NULL,

`event` varchar(20) DEFAULT NULL,

`delay` int(11) DEFAULT NULL,

PRIMARY KEY (`tname`,`aname`,`groupOrder`),

CONSTRAINT `grouping\_ibfk\_1` FOREIGN KEY (`tname`, `aname`) REFERENCES `activity` (`tname`, `aname`) ON DELETE CASCADE

)

CREATE TABLE `groupingDetails` (

`tname` varchar(30) DEFAULT NULL,

`aname` varchar(30) DEFAULT NULL,

`groupOrder` int(11) DEFAULT NULL,

`cid` varchar(5) DEFAULT NULL,

`animation` varchar(40) DEFAULT NULL,

`cOrder` int(11) DEFAULT NULL,

`duration` int(11) DEFAULT NULL,

KEY `tname` (`tname`,`aname`,`groupOrder`),

KEY `tname\_2` (`tname`,`aname`,`cid`),

CONSTRAINT `groupingDetails\_ibfk\_1` FOREIGN KEY (`tname`, `aname`, `groupOrder`) REFERENCES `grouping` (`tname`, `aname`, `groupOrder`) ON DELETE CASCADE,

CONSTRAINT `groupingDetails\_ibfk\_2` FOREIGN KEY (`tname`, `aname`, `cid`) REFERENCES `content` (`tname`, `aname`, `cid`) ON DELETE CASCADE

)

CREATE TABLE `hints` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`cid` varchar(5) NOT NULL,

`bigHint` text,

`smallHint` text,

`url` text,

PRIMARY KEY (`tname`,`aname`,`cid`),

CONSTRAINT `hints\_ibfk\_1` FOREIGN KEY (`tname`, `aname`, `cid`) REFERENCES `content` (`tname`, `aname`, `cid`) ON DELETE CASCADE

)

CREATE TABLE `text` (

`tname` varchar(30) NOT NULL,

`aname` varchar(30) NOT NULL,

`cid` varchar(5) NOT NULL,

`data` text,

PRIMARY KEY (`tname`,`aname`,`cid`),

CONSTRAINT `text\_ibfk\_1` FOREIGN KEY (`tname`, `aname`, `cid`) REFERENCES `content` (`tname`, `aname`, `cid`) ON DELETE CASCADE

**)**

**BACKEND**

The backend ensures that an instance of an activity is stored as it is being passed from the editor’s view. The request is processed by the servlet StoreActivity which converts the json passed into an in memory data structure.

This data structure corresponds to the data structure used by the front end.

Similary the database design also corresponds to the in memory data structures being used throughout the project.

The ER diagram and the schema DDL definition resembles the above claim.

The technology used to pass data to the backend is AJAX. This is used so that while requesting an activity either on the student side or the teacher’s side, the page refresh is not required, hence making the activity fetch procedure less bulky.

The database management system used in this project is MYSQL.

As the request for an activity is received either from the teaxher’s side or the student’s side, the data for that specific activity is loaded in the in memory datastructure and then it is converted in the json file by the Converter class.

The controller for these activity is the RetrieveActivity servlet.

The project follows the MVC Architecture.

Package datastore is the link to the database.

Package data model contains all the model classes which are required and resembles different components of an activity.

Package controller consists of the servlets which manage the flow of the data.

When an write for an activity is requested, the servlet StoreActivity fiirst checks if a topic is available with the topic name being passed. If not, it creates a new topic.

If the activity is already present and hence needs to be overwritten delete command for that activity is first sent to flush all the data about that activity. Thereafter the activity is stored freshly. This is done keeping in mind that in the worst case all the previous elements of an activity can be deleted and a complete new set of data might be available. Moreover this approach simplifies the flow of the application to a great extent as the number of elements in the activity is quite large and checking if each exists for update queries is not performance efficient. To provid for the safeguard of the data , the delete and insert operations are wrapped into a single transaction so that at no cost the data is lost.

LearnerView servlet is the controller which redirects to either the student view or the teacher’s view depending on the url value passed by the UserType.jsp file. It is also responsible for loading the topic tree on both the sides which is also the main reason for providing an interface for the teacher’s view and the student’s view instead of directly redirecting to them from the UserType.jsp file.

**TECHNOLOGIES USED**

JavaScript Libraries:

MathJax

Fabric.js

CKEditor

HTML / CSS

For front-end purpose, HTML / CSS were used.

JavaScript

Other than the libraries, JavaScript has also been extensively used to control the flow of simulations.

JSP

We used JavaServer Pages (JSP) to communicate with the server.

Learner’s requirements

The choice of above technologies has made our tools compatible across multiple operating systems (Windows, MacOS, Linux, Android, etc) and multiple devices (laptops, mobiles, tablets). The only software needed to run our simulations is a web browser with html canvas support (most modern browsers: Firefox, Chrome, Safari).

For implementing these shapes for the graphics portion we have used a javascript library fabric.js. Fabric.js is simple and powerful javascript HTML5 canvas library.

**Fabric.js**

[**Fabric.js**](http://fabricjs.com/) **— a powerful Javascript library that makes working with HTML5 canvas a breeze. Fabric provides a missing object model for canvas, as well as an SVG parser, layer of interactivity, and a whole suite of other indispensable tools. It is a fully open-source project, licensed under MIT, with many contributions over the years.**

**I started developing Fabric 3 years ago, after discovering the pains of working with native canvas API. I was creating an interactive design editor for** [**printio.ru**](http://printio.ru/) **— my startup that allows users to design their own apparel. The kind of interactivity we wanted only existed in Flash apps in those days. Even right now, very few come close to what became possible with Fabric.**

[**Canvas**](http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html) **allows us to create some** [**absolutely**](http://net.tutsplus.com/articles/web-roundups/21-ridiculously-impressive-html5-canvas-experiments/)[**amazing**](http://speckyboy.com/2011/12/07/20-amazing-implementations-of-html5-canvas/)[**graphics**](http://artatm.com/2012/01/23-truly-amazing-and-unbelievable-html5-canvas-and-javascript-experiments/) **on the web these days. But the API it provides is** [**disappointingly low-level**](http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html#2dcontext)**. It's one thing if we simply want to draw few basic shapes on canvas and forget about them. But as soon as there's need for any kind of interaction, change of picture at any point, or drawing of more complex shapes — situtation changes dramatically.**

**Fabric aims to solve this problem.**

**Native canvas methods only allow us to fire off simple graphic commands, blindly modifying entire canvas bitmap. Want to draw a rectangle? Use fillRect(left, top, width, height). Want to draw a line? Use a combination of moveTo(left, top) and lineTo(x, y). It's as if we're painting canvas with a brush, layering more and more oil on top, with very little control.**

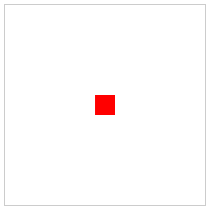
**Instead of operating on such low level, Fabric provides simple but powerful object model on top of native methods. It takes care of canvas state and rendering, and lets us work with “objects” directly.**

**Let's take a look at a simple example demonstrating this difference. Let's say we want to draw a red rectangle somewhere on canvas. Here's how we would do this with native <canvas> API.**

**// reference canvas element (with id="c")  
var canvasEl = document.getElementById('c');  
  
// get 2d context to draw on (the "bitmap" mentioned earlier)  
var ctx = canvasEl.getContext('2d');  
  
// set fill color of context  
ctx.fillStyle = 'red';  
  
// create rectangle at a 100,100 point, with 20x20 dimensions  
ctx.fillRect(100, 100, 20, 20);**

**Now, let's take a look at doing the same with Fabric:**

**// create a wrapper around native canvas element (with id="c")  
var canvas = new fabric.Canvas('c');  
  
// create a rectangle object  
var rect = new fabric.Rect({  
 left: 100,  
 top: 100,  
 fill: 'red',  
 width: 20,  
 height: 20  
});  
  
// "add" rectangle onto canvas  
canvas.add(rect);**



**At this point, there's almost no difference in size — the two examples are pretty similar. However, you can already see how different the approach to working with canvas is. With native methods, we operate on context — an object representing entire canvas bitmap. In Fabric, we operate on objects — instantiate them, change their properties, and add them to canvas. You can see that these objects are first-class citizens in Fabric land.**

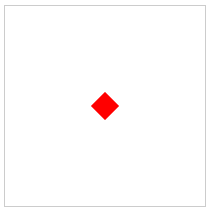
**But rendering plain red rectangle is just so boring. We could have at least made something fun with it! Perhaps, rotate slightly?**

**Let's try 45 degrees. First, using native <canvas> methods:**

**var canvasEl = document.getElementById('c');  
var ctx = canvasEl.getContext('2d');  
ctx.fillStyle = 'red';  
  
ctx.translate(100, 100);  
ctx.rotate(Math.PI / 180 \* 45);  
ctx.fillRect(-10, -10, 20, 20);**

**and now using Fabric:**

**var canvas = new fabric.Canvas('c');  
  
// create a rectangle with angle=45  
var rect = new fabric.Rect({  
 left: 100,  
 top: 100,  
 fill: 'red',  
 width: 20,  
 height: 20,  
 angle: 45  
});  
  
canvas.add(rect);**



**What happened here?**

**All we had to do in Fabric is change object's “angle” value to 45. With native methods however, things become much more “fun”. Remember that we can't operate on objects. Instead, we tweak positioning and angle of entire canvas bitmap (ctx.translate, ctx.rotate) to suit our needs. We then draw rectangle again, but remembering to offset bitmap properly (-10, -10), so that it's still rendered at the point of 100,100. As a bonus exercise, we had to translate degrees to radians when rotating canvas bitmap.**

**I'm sure you're starting to see exactly why Fabric exists and how much low-level boilerplate it hides.**

**But let's take a look at yet another example — keeping track of canvas state.**

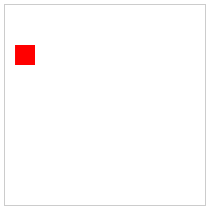
**What if at some point, we wanted to move that now-familiar red rectangle to a slightly different location on canvas? How would we do this without being able to operate on objects? Would we just call another fillRect on a canvas bitmap?**

**Not quite. Calling another fillRect command actually draws rectangle on top of whatever is already drawn on canvas. Remember I mentioned painting with a brush earlier? In order to “move” it, we'd need to first erase previously drawn content, and then draw rectangle at a new location.**

**var canvasEl = document.getElementById('c');  
  
...  
ctx.strokRect(100, 100, 20, 20);  
...  
  
// erase entire canvas area  
ctx.clearRect(0, 0, canvasEl.width, canvasEl.height);  
ctx.fillRect(20, 50, 20, 20);**

**And how would we accomplish this with Fabric?**

**var canvas = new fabric.Canvas('c');  
...  
canvas.add(rect);  
...  
  
rect.set({ left: 20, top: 50 });  
canvas.renderAll();**



**Notice a very important difference. With Fabric, we no longer need to erase the content before attempting to “modify” any content. We still work with objects, simply changing their properties, and then re-render canvas to get a “fresh picture”.**

### **Objects**

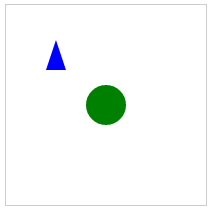
**We already saw how to work with rectangles by instantiating fabric.Rect constructor. But of course Fabric covers all the other basic shapes as well — circles, triangles, ellipses, and so on. All of them are exposed under fabric “namespace” as fabric.Circle, fabric.Triangle, fabric.Ellipse, etc.**

**7 basic shapes provided in Fabric:**

* [**fabric.Circle**](http://fabricjs.com/docs/fabric.Circle.html)
* [**fabric.Ellipse**](http://fabricjs.com/docs/fabric.Ellipse.html)
* [**fabric.Line**](http://fabricjs.com/docs/fabric.Line.html)
* [**fabric.Polygon**](http://fabricjs.com/docs/fabric.Polygon.html)
* [**fabric.Polyline**](http://fabricjs.com/docs/fabric.Polyline.html)
* [**fabric.Rect**](http://fabricjs.com/docs/fabric.Rect.html)
* [**fabric.Triangle**](http://fabricjs.com/docs/fabric.Triangle.html)

**Want to draw a circle? Just create a circle object, and add it to canvas. Same with any other basic shape:**

**var circle = new fabric.Circle({  
 radius: 20, fill: 'green', left: 100, top: 100  
});  
var triangle = new fabric.Triangle({  
 width: 20, height: 30, fill: 'blue', left: 50, top: 50  
});  
  
canvas.add(circle, triangle);**



**..and there we have a green circle, drawn at 100,100 location and a blue triangle at 50,50 location.**

# CKEditor

CKEditor is the tool we are using to support text formatting in CUBENTS. **Inline Editing form CKeditor is incorporated in CUBENTS.**

**Inline Editing is a new technology that allows you to select any editable element on the page and edit it in-place. As a result, the editor can be used to edit content that looks just like the final page. It is a total WYSIWYG experience, because not only the edited content looks like the final outcome, but also the page and the context where the content is placed is the real one. Unlike in classic editor, there is no <iframe> element created for the editing area. The CSS styles used for editor content are exactly the same as on the target page where this content is rendered!**

It comes with following inbuilt features:

#### Advanced text formatting

#### From basic formatting to applying pre-defined text formats, text styles and creating mathematical formulas or beautiful code snippets

## **Fully accessible**

**CKEditor complies with the latest web accessibility standards (WCAG 2.0, Section 508, WAI-ARIA) and enables creation of accessible content thanks to a built-in Accessibility Checker.**

## **Spell Check As You Type**

**See and correct misspellings while typing. Misspelled words are underlined and you can simply select a suggestion to replace the incorrect word**

## **Clean code**

**CKEditor creates code that is clean, standards-compliant and readable. You can also include your custom formatting rules to fully control your HTML output.**

## **Ease of integration**

**Integrate CKEditor instantly by adding one script to your page. Choose between the ready-made zip installation package, your favorite package manager or the CDN version**

## **Fully customizable**

**Customize every single aspect of CKEditor (i.e. toolbar, skin, color palette, dialogs, context menu, data parsing, styling, etc.) and adjust it to your needs.**

## **Rich API**

**Powerful and extensive API lets you interact with the editor at runtime, create custom features and manipulate content at will.**

**Code for including CDN of inline CKeditor**

**<script src="**[**https://cdn.ckeditor.com/4.7.0/full/ckeditor.js**](https://cdn.ckeditor.com/4.7.0/full/ckeditor.js)**"></script>**

**Code for adding inline CKeditor**

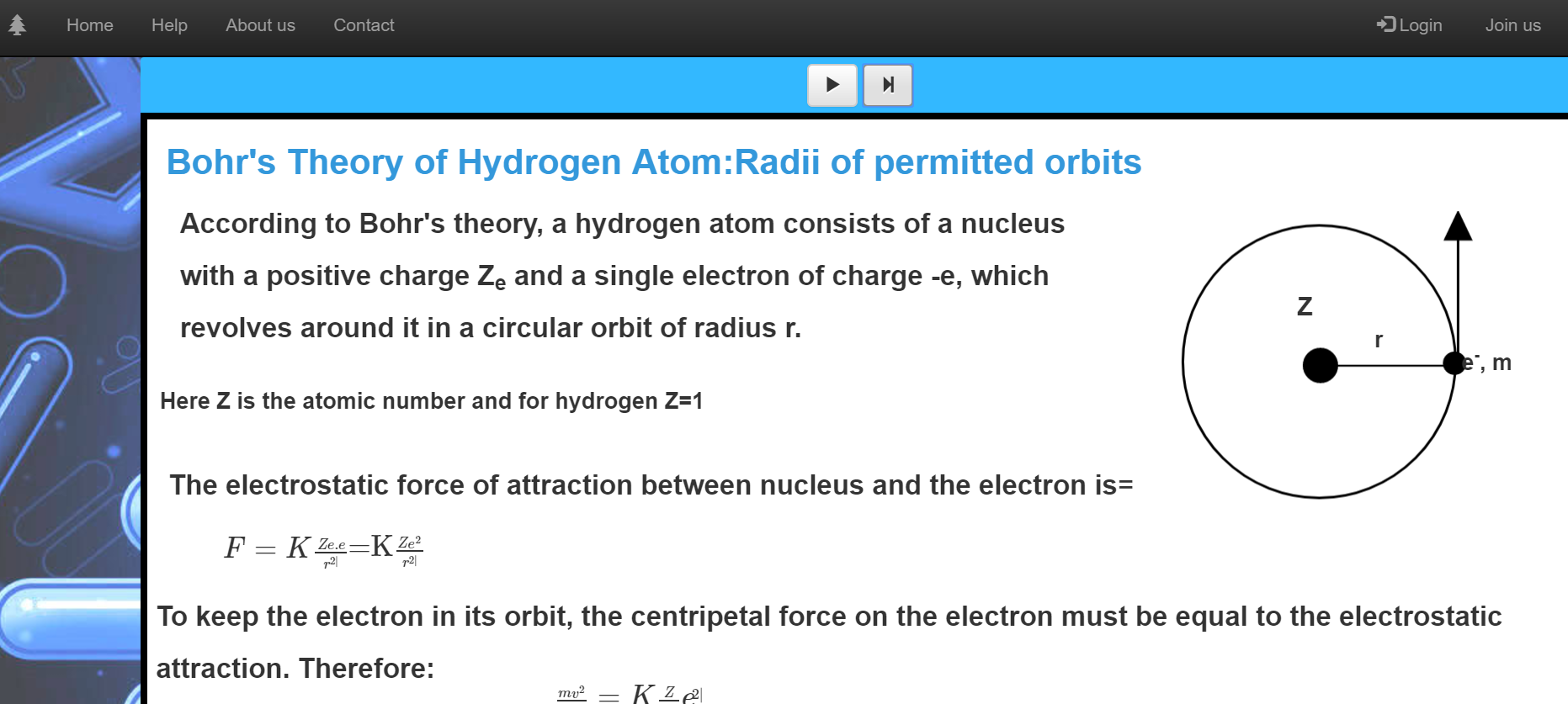
**<script>**

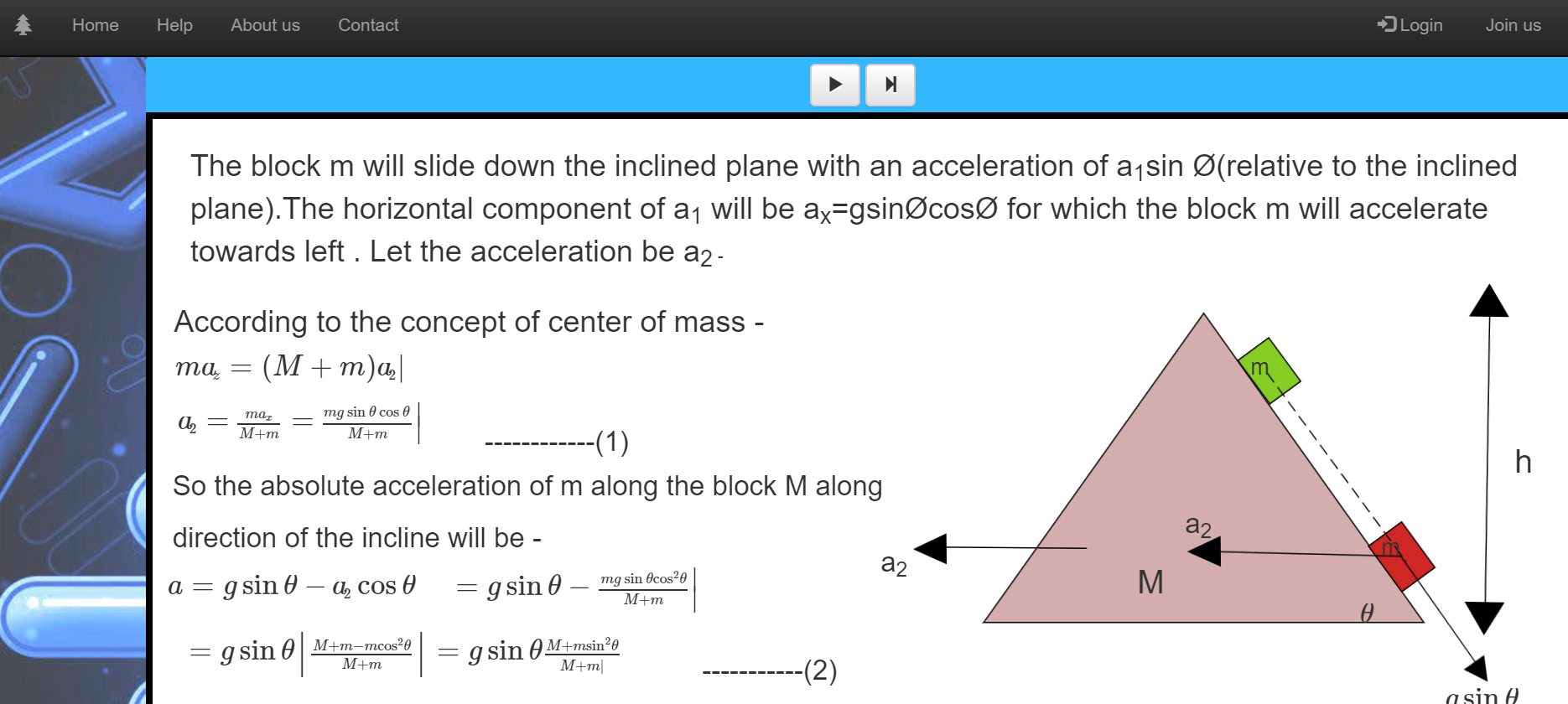
**CKEDITOR.inline( "T"+c,{});**

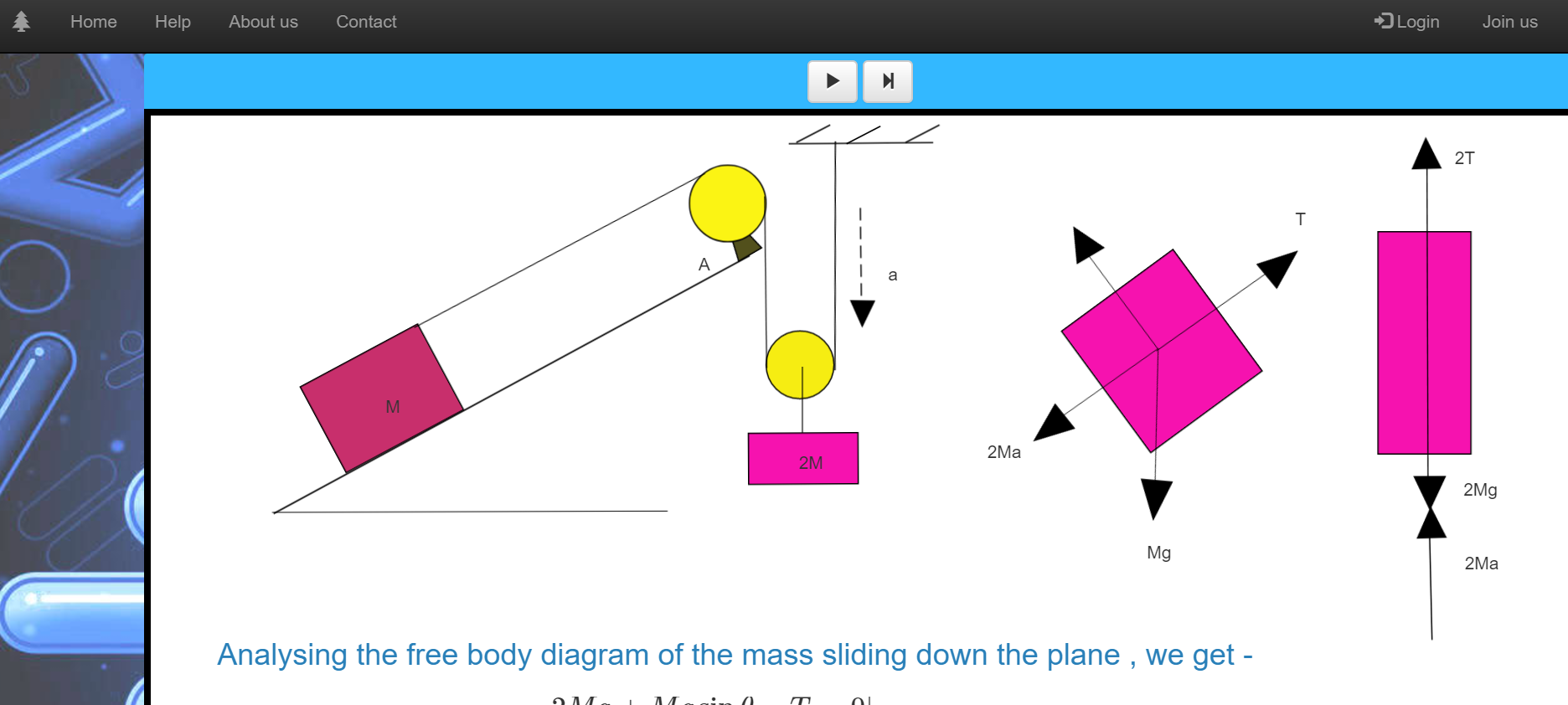
**</script>**

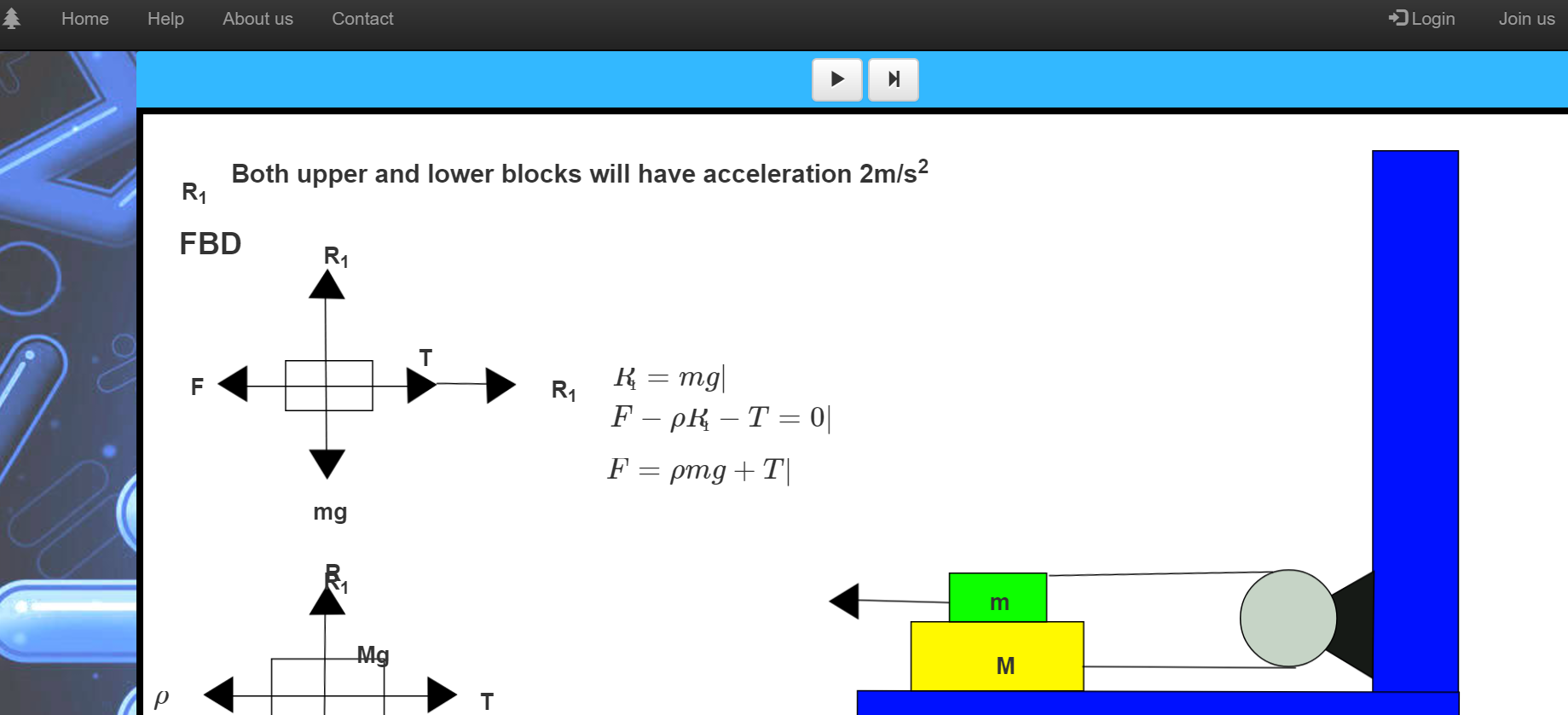
**In the above code an inline ckeditor instance is associated with a div that is having id T0, T1, and so on depending on the value of ‘c’.**

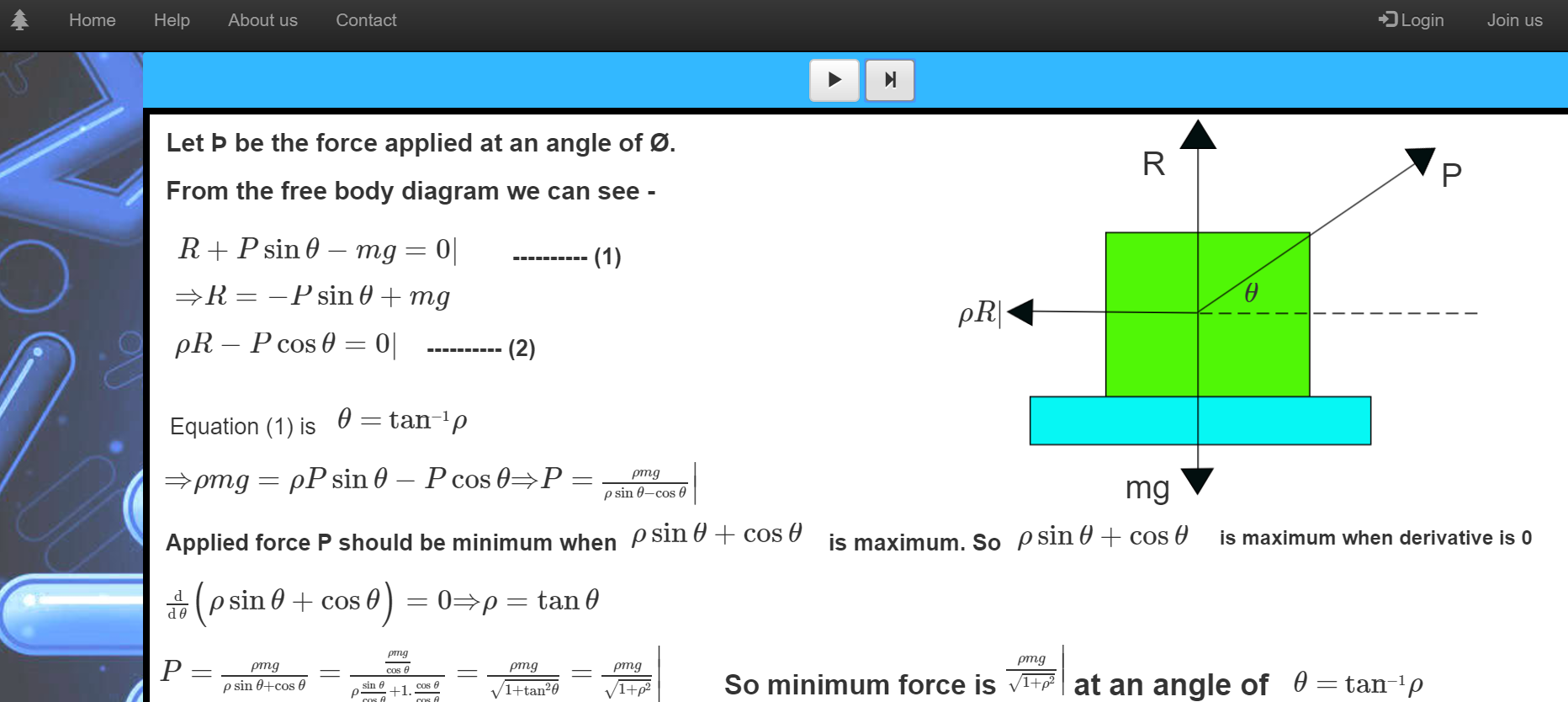
**DEMONSTRATIONS**

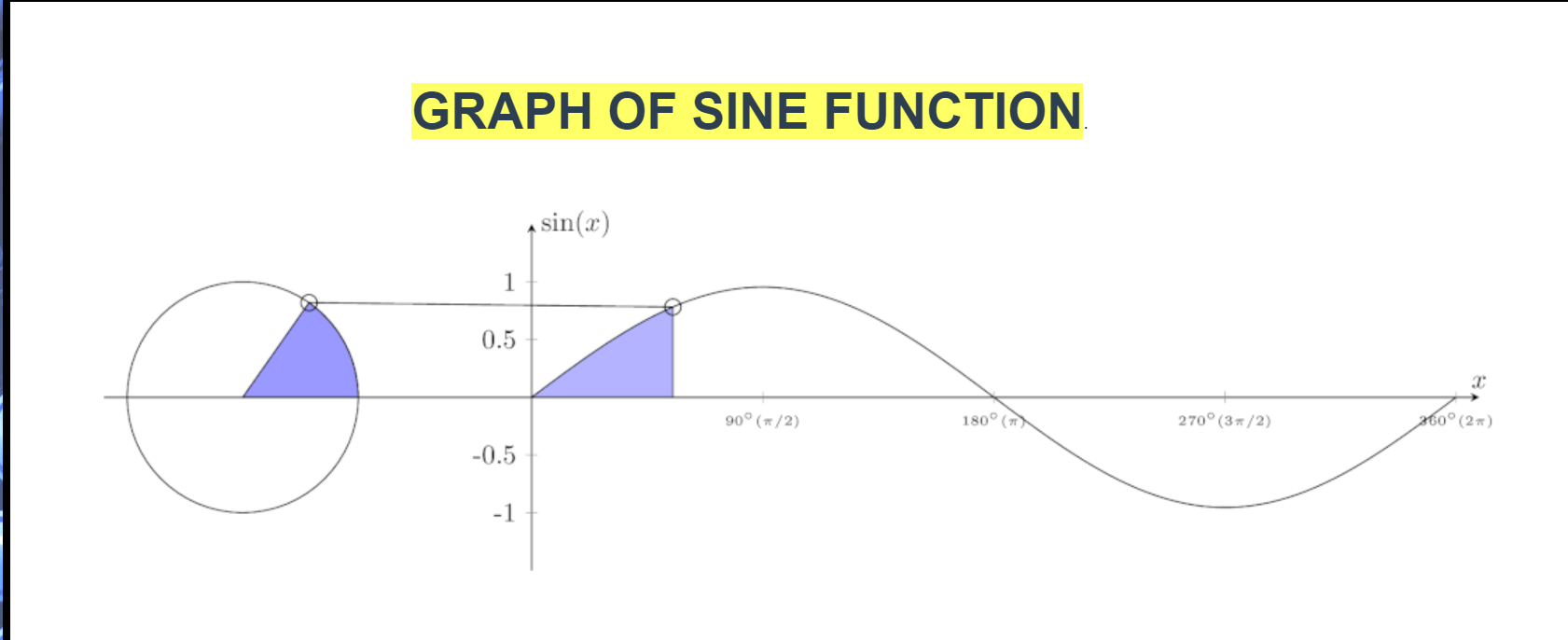


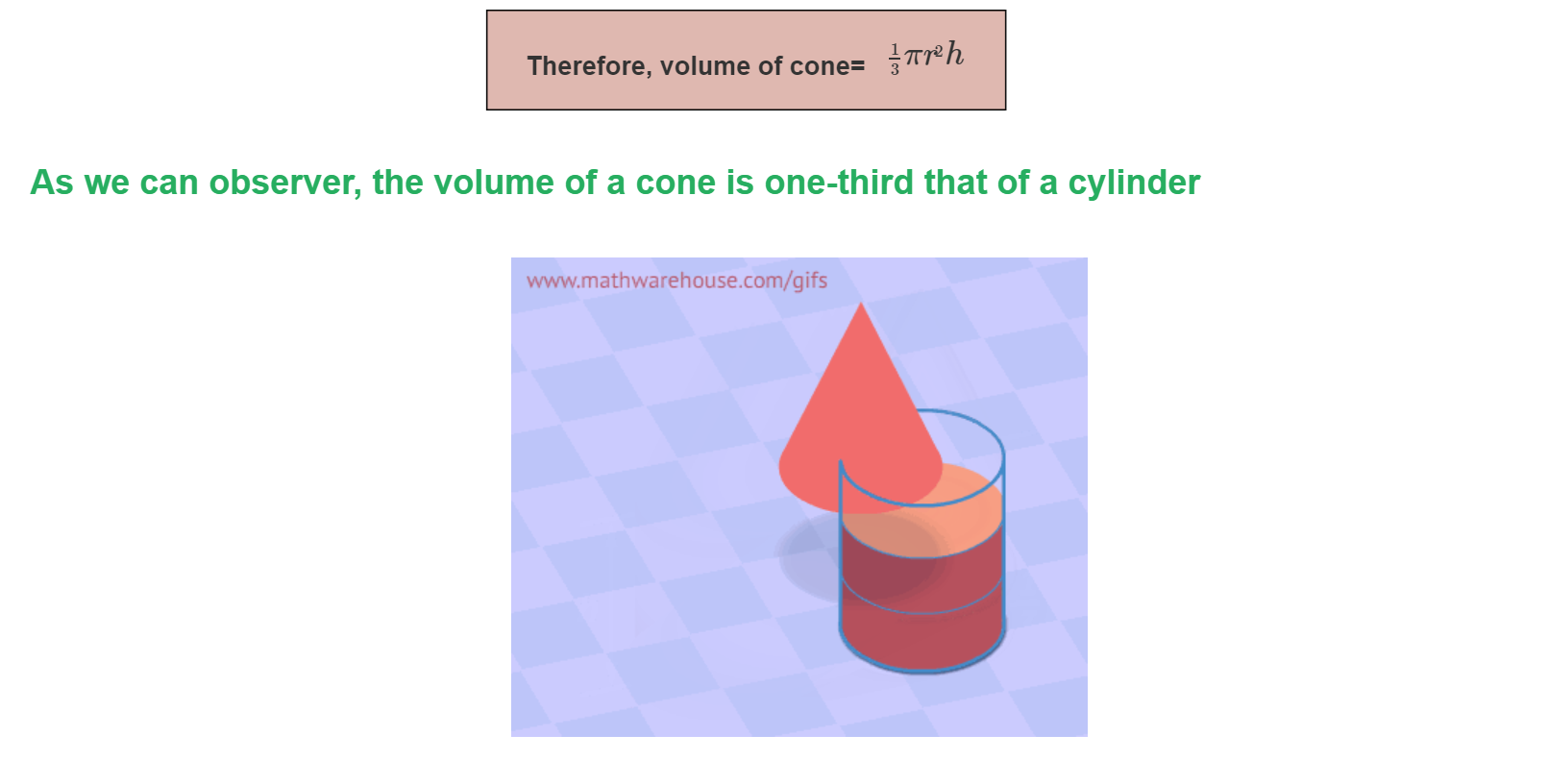


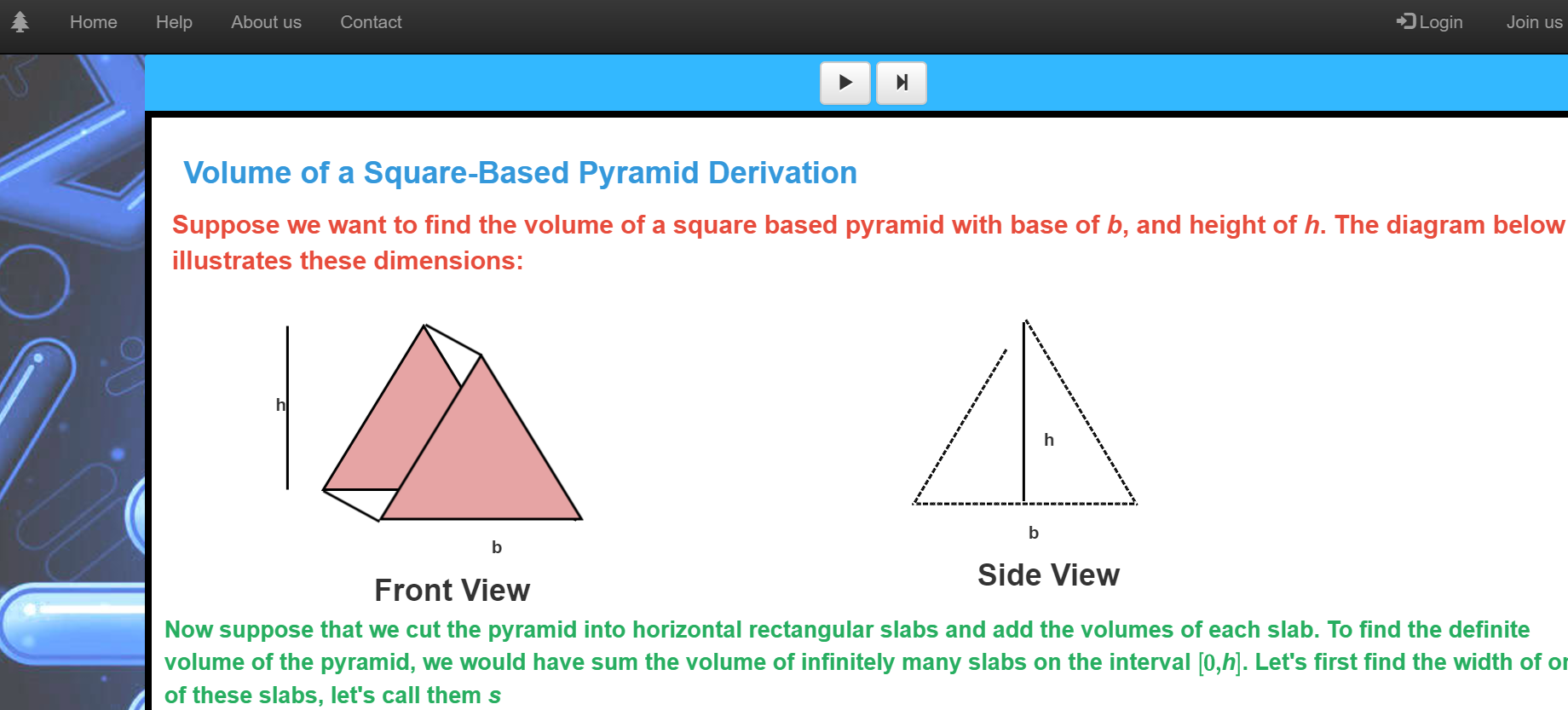












**LEARNINGS**

We have learned to develop a software on industrial standards by adhering to the specifications of the program. We were fortunate to experience this idea to implementation journey before even stepping into the industry. We learned how to apply the object orientation principles for making code reusable. This is a dire requirement for any programmer to work in an industry where reusability and efficiency plays an important role.

In the words of Phil Jackson- “The strength of the team is each individual member. The strength of each member is the team.” Keeping true to his words, we had a team of 7 members and each one worked to his full capacity to make this project a success. Working in a team was a huge challenge as everyone had different ideas and viewpoint. So keeping this in consideration we had to find out solutions so that output could be maximized.We learned the depths of HTML and CSS. We had a wide exposure to the usage of Javascript libraries and how to extend these to suit our requirements. We had to work under strict guidelines with desired specifications to be completed which led us to develop many soft skills such as professionalism, discipline, working under deadlines and constraints.

**CHALLENGES**

1. Building the equation parser

2. Integration of different element layers.

3. Ensuring compatibility across devices and browsers cross-­browser compatibility was important for us so as to ensure that the platform is never a restriction for the learner.

4. Making the application user friendly. We put our utmost effort to ensure that the user could easily start creating content on the application.

**CONCLUSION**

With all the features that this equation editor provides, and also provided that its not just equations but equations mingled with graphics on the canvas that makes it unique on the internet and makes it an invaluable tool for creating tutorials and mathematical/scientific presentations.

**BIBLIOGRAPHY**