Google Summer of Code

Upgrade third-party libraries

Google Summer Of Code 2019

Nisheal A. John (@Nishealj)

Karunya University

nisheaalajohn@karunya.edu.in

*Hi, I'm **Nisheal A. John** a pre-final year student at Karunya University Coimbatore pursuing my bachelor's degree in Computer Science and Engineering. I have always manifested a strong academic record and I've the zeal of learning, i never failed to miss the opportunity to learn more.*

Thank you for giving me an opportunity to contribute to your organization through GSoC, I've good working experience in software development and looking forward to work and grow with oppia.

Why are you interested in working with Oppia?

I'm Contributing to oppia from a quite a few months ago and oppia is full of opportunities. The work experience and skills gained in this open source community is immeasurable.The educational concept which oppia is building is much needed and this is bringing smiles to millions of students across the globe. The oppia community is delightful and always ready to help, it's great to learn and grow with such a welcoming organization I'd love to see my future in oppia!

What interests you about this project? Why is it worth doing?

I've good working experience in software development and testing, I've been working with python frameworks and AngularJS from a quite long. The oppia codebase is built on the same and that is helping me to grow faster. I have always worked well with the infrastructural management and debugging, the project i've chosen is also an infrastructural upgrade it needs much of testing and that excites me more. I have good knowledge about the needed libraries in a codebase making me an ideal asset to go ahead with this project as it is easier for me identify the use case and importance of the libraries used in oppia.

Upgrading the third party libraries will help oppia towards its infrastructural upgrade and an upgraded infrastructure will help in building remarkable features in future.

Making this project a worth doing.

Prior Experience :

I've hosted various projects in python framework and angularJS, ***SEN10S*** *is one of the notable projects among those :*

- ***SEN10S*** *is software as a service for blog writers and readers. This web Application is built using the cutting edge technologies like Django, AngularJS and nodeJS for its cellular response. It also provide data analytics using R .*

I've also worked with :

- ***3SC SUPPLY CHAIN PVT LTD,*** *in Pune where my work was to build and deploy supply chain management softwares using Python framework.*

Contributions to oppia :

I've notable contribution in the bug-fixing as a team member,
PRs for bug fix and improvements

- #6435 : *CSRF_CHECK Flag Flipped in Error 404 Handler class.*
- #6463 : *Changing error status for missing csrf_token from 500 to 401.*
- #6498 : *Added regex to validate return_url in signup page*
- #6532 : *Fixes issues while selecting an answer in the editor feedback tab.*
- #6544 : Fixes Strange dragging behaviour of play later list in learner dashboard.
- #6547 : *Reordering Hints and Worked Examples in Safari*

Here is the complete list of PRs raised by me.

Project Overview:

Upgrade third-party libraries

This project focuses on upgrading all the third party libraries which oppia uses into its newer version so that we might be able to solve issues in the issue tracker. Upgrading third-party libraries will give us speed and feature improvements and will also help us in easy migration to further cutting edge infrastructures later.

Libraries in ./third_party :

Library	Current Version	Latest Version
backports.functools_lru_cache	1.5	1.5
Bleach	1.2.2	3.1.0
gaeCloudStorage	1.9.15.0	1.9.22.1
gaeMapReduce	1.9.17.0	1.9.84
gaePipeline	1.9.17.0	1.9.84
graphy	1.0.0	1.0.0
html5Lib	0.95	1.0.1
requests	2.10.0	2.21.0
simpleJson	3.7.1	3.16.1
beautifulSoup	4.7.1	4.7.1
mutagen	1.38.0	1.43.0
soupsieve	1.8	1.9.0
fontAwesome	4.7.0	5.8.1

Static Libraries in ./third_party/static :

Library	Current Version	Latest Version
angularAudio	f403ea9e86de141278d410bade8ef1f43e6d188	1.7.3
angularRecorder	1.4.1	1.4.1
angularTranslate	2.8.1	2.18.1
angularCookiesRev	1.4.0	1.7.8
angularDragAndDrop	2.1.0	2.1.0
angularStorageCookiesRev	2.8.1	2.18.1
messageFormat	0.3.1	2.0.5
angularTranslateInterpolationMessageFormat	2.11.0	2.18.0
angularToastr	1.7.0	2.1.1
bootstrap	3.3.4	4.3
codemirror	3.19.0	5.45.0
leaflet	1.4.0	1.4.0
guppy	b5055b963fdbea5c6c1e92dbf58fdaf3ea0cd8ba	2.0.0
hammerJs	2.0.4bo	2.0.8
headroomJs	0.9.4	0.9.4
waveSurferJs	1.0.46	2.2.1
jquery	3.2.1	3.3.1
jqueryUI	1.12.1	1.12.1
mathExpressions	1.7.0	2.0.0.alpha
mathJax	2.6.0	2.7.0
mathJs	4.2.2	5.8.0
midiJs	2ef687b47e5f478f1506b47238f3785d9ea8bd25	0.3.0
ngJoyride	5b25d076405c9a787db602d2a8ff49919228f4fc	0.1.20
ngImageCrop	0.3.2	-
ngInfiniteScroll	1.0.0	1.0.0
select2	4.0.3	4.0.6
ckeditor	4.9.2	Ckeditor 5 : 12.0.0

ckeditorSharedspace	4.9.2	4.11.3
ckeditorBootstrapCK	1	1
uiBootstrap	2.5.0	2.5.0
uiCodemirror	0.1.2	0.3.0
uiSortable	0.17.1	0.19.0
uiUtils	0.1.1	3.0.0
yuiCompressor	2.4.8	2.4.8
browsermob-proxy	2.1.1	2.4.1

Analysis of these libraries based on :

1. **Priority :**

   The libraries should be prioritized. There are some libraries for which upgrading them will fetch us speed enhancement immediately whereas for some libraries their upgrade maynot be noticeable directly.

2. **ChangeLogs:**

   The changelogs of all the libraries should be checked for new features which can be added and the bugs which are fixed in the libraries which can help us to solve issues in our issue tracker.

3. **Support Changes needed :**

   The newer version of some libraries can be dependent on upgrades of other libraries also. We need to clearly identify the support libraries and upgrade them carefully. Also the library may have changed or renamed their methods in the newer version we need to updated such methods in our codebase

4. **Backward Compatibility :**

   The newer version the libraries should be interoperable with the codebase, many library in their newer update are not providing python 2.x support whereas the oppia uses python 2.x we need to identify such libraries and find an alternative.

<table>
<tr><th></th><th>Library</th><th>Changelog & Support Changes</th></tr>
<tr><td>1.</td><td>FontAwesome
(4.7.0 to 5.8.1)</td><td>
Changelog :

Introduced 2 Different Free Icon Styles solid, regular, and light
fa class should be removed and new class far fab or fal should be added and
Font folder has been renamed to webfonts

Support Changes :

**Common Changes needed
Python script to remove of ‘fa’ class and write fab or far.
All the fa-<icon name> should be changed to far fa-<icon name> Example fa fa-facebook is not supported must be changed to fab fa-facebook.

<table>
<tr><th>Icon Style</th><th>Prefix</th><th>Example</th></tr>
<tr><td>Font Awesome Brands</td><td>fab</td><td>fab fa-facebook</td></tr>
<tr><td>Font Awesome Solid</td><td>fas or fa</td><td>fas fa-facebook</td></tr>
</table>
Tests Needed :

**Common tests
Manual testing to verify logo and button icons

We use Font-Awesome icons in many views which are outdated now, upgrading Font-Awesome will fetch us new logos and button.
<table>
<tr><th>Font-awesome 4</th><th>Font-awesome 5</th></tr>
<tr><td>  </td><td>  </td></tr>
</table>
</td></tr>
<tr><td>2.</td><td>Bleach
(1.2.3 to 3.1.0)</td><td>
Changelog :

Bleach no longer depends on html5lib

</td></tr>
</table>

	Bleach is an allowed-list-based HTML sanitizing library that escapes or strips markup and attributes.	• *Bleach.clean() no longer un escapes entities* • *Dropped support for Python 3.3* • *Function names updated* *Support Changes :* • *Bleach.clean() in html_cleaner.py Should be rewritten.* • *Other depreciation functions and parameter should be rewritten.* *Tests Needed :* • *Running Frontend and Backend tests for breaking changes*
3.	**gaeCloudStorage (1.9.15.0 to 1.9.22.1)**	*Support Changes :* • *Version change in manifest.json* *Tests Needed :* • *Running Frontend and Backend tests for breaking changes*
4.	**html5Lib (0.95 to 1.01)** *html5lib is a pure-python library for parsing HTML widely used with beautiful soup for parsing*	*Changelog :* • *Python 3.2+ supported* • *Add support for setuptools < 18.5 to support environment markers* *Support Changes :* • *Version change in manifest.json* *Tests Needed :* • *Running Frontend and Backend tests for breaking changes*
5.	**Requests (2.10.0 to 2.21.0)** *Requests is a HTTP library for*	*Changelog :* • *Python 2.6 is not supported* • *Removal of the 301 redirect cache. This improves thread-safety* • *Memory Enhancements*

	Python.	*Support Changes :* • *Version change in manifest.json* *Tests Needed :* • *Running Frontend and Backend tests for breaking changes*
6.	**simpleJson** **(** 3.7.1 to 3.16.1 **)** *simplejson is a fast and extensible JSON encoder and decoder for Python.*	*Changelog :* • *Python 3 support* • *simplejson is now available as wheels.* • *Better error handling* • *strings with speedups enabled on Python 2.x* *Support Changes :* • *Version change in Gae_suite.py, Appengine_config.py manifest.json* *Tests Needed :* • *Running Frontend and Backend tests for breaking changes*
7.	**Mutagen** **(** 1.38.0 **to** 1.43.0 **)** *Mutagen is a Python module to handle audio metadata, used for audio / audio-translation in oppia.*	*Changelog :* • *Removed support for Python 3.4* • *MP3: Improved bitrate accuracy for files with XING header* • *Bugs Fixed* *Support Changes Need : None* *Tests Needed :* • *Running Frontend and Backend tests for breaking changes* • *Manual testing in audio translation*
8.	**Soupsieve** **(** 1.8.0 **to** 1.9.0 **)** *Soup Sieve is a CSS selector*	*Changelog :* • *Bugs Fixed.* • *Python 3x+ support.* *Support Changes Need : None*

	library designed to support Beautiful Soup 4	*Tests Needed :* ● *Backend & Frontend Tests*
9.	**messageFormat**	*Changelog :* ● *Drop support for #compile().toString('exports')* ● *Replace addLocale with addMessages* *Support Changes Need : None*
11.	**Codemirror** **(** 3.19.0 **to** 5.54.0 **)** *CodeMirror is a versatile text editor implemented in JavaScript for the browser*	*Changelog :* ● *Bugs Fixed.* ● *Improved Rendering in safari* *Support Changes Need :* *Tests Needed :* ● *Frontend Tests.* ● *Manual Testing with Morden browsers*
12.	**hammerJs** **(** 2.0.4 **to** 2.0.8 **)** *Hammer is a javascript library that can recognize gestures made by touch, mouse and pointer Events*	*Changelog :* ● *Bugs fixed* ● *Now Compatible with requirejs optimizer namespaces* *Support Changes Need : None*
13.	**waveSurferJs** **(** 1.0.46 **to** 2.2.1 **)**	*Changelog :* ● *Safari support*

	Customizable audio waveform with visualization	
14.	**Jquery** **(** 3.2.1 **to** 3.3.1**)**	*Changelog :* ● *Deprecated jQuery.now jQuery.isWindow jQuery.isFunction and more* *Support Changes Need :*
15.	**mathExpressions** **(** 1.7.0 **to** 2.0.0 **)** Perform basic equality testing and symbolic computations on mathematical expressions	*Changelog :* *Support Changes Need :* ● *Only version changes*
16.	**uiCodemirror** **(** 0.1.2 **to** 0.3.0 **)** *Supports codemirror of textareas*	*Changelog :* ● *onChange event is now handled by ngChange.* ● *Supports CodeMirror 5*
17.	**mathJs** **(** 4.2.2 **to** 5.8.0**)** *Mathematical expression parser with support for symbolic computation.*	*Changelog :* ● *Implemented functions equalText and compareText* ● *More consistent behavior of sqrt, nthRoot, and pow* ● *Removed instances of Number, Boolean, and String from clone and typeof* ● *Function rename and rewritten* *Support Changes Need :* ● *Only version changes*
18.	**midiJs**	*Changelog :*

	(to 0.3.0 **)** *Supports oppia in creating MIDI apps*	• *Bug Fix* *Support Changes Need :* • *Only version changes*
19.	**Select2** **(** 4.0.3 to 4.0.6 **)** *The jQuery replacement for select boxes*	*Changelog :* • *Fixed browser related issues (Safari , IE)* • *Utils.__cache instead of using changed $.data* • *Bug Fix* *Support Changes Need :* • *Only version changes*
20.	**Browsermob-proxy** **(** 2.1.1 **to** 2.4.1 **)** *watch and manipulate network traffic from AJAX Requests*	*Changelog :* • *Removed deprecated uadetector* • *Simplified starter scripts* *Support Changes Need :* • *Only version changes*
21.	**Ckeditor Sharedspace** **(** 4.9.2 **to** 4.11.3**)** *Helps in sharing the editor toolbar*	*Changelog : None* *Support Changes Need : None*
22.	**Bootstrap** **(v3 to v4)**	*Changelog :* • *In v4, Many class name and styles are rewritten or removed. mostly with the components like button, navigation, card and forms* • *The primary unit changed from px to em* *Support Changes Need :* • *A Task will be created using gulp.task to update all the renamed classes at once*

Salient Upgrades in third_party libraries :

I believe Upgrading the below mentioned libraries are very important and that's why they are termed as the salient upgrade. Upgrading these libraries will have a direct impact on the performance and will us to fix many issues in the issue tracker.

1. **CkEditor4 to CKEditor 5 (** Issues **:** #6225 **)**
2. **ui-Sortable & jQuery Upgrade (** Relatable *issues* **:** #3819 #6229 **)**
3. **MathJax to 2.7.0 or KaTex (** Relatable issue **:** #2506 **)**
4. **Upgrading GAE (** gae.pipeline, google-sdk, gaeCloudStorage **)**
5. **Migrating to Font Awesome 5**
6. **Migrating to Bleach 3 with html5lib**
7. **Upgrading bower based Angular static libraries**
8. **Migrating to BootStrap 4**

1. Migration from CKEditor to CKEditor 5

CKEditor is a WYSIWYG rich text editor which enables writing content directly inside of web pages or online applications. CKEditor4 is widely used in oppia to create/edit exploration. CKEditor play significant role in enhancing the creator experience.

*When compared CKEditor4 to CKEditor5 it is considered a totally new editor in terms of **Design**, **integration**, **features data mode**l and **API.** Below i've covered the technical design and approach to this migration*

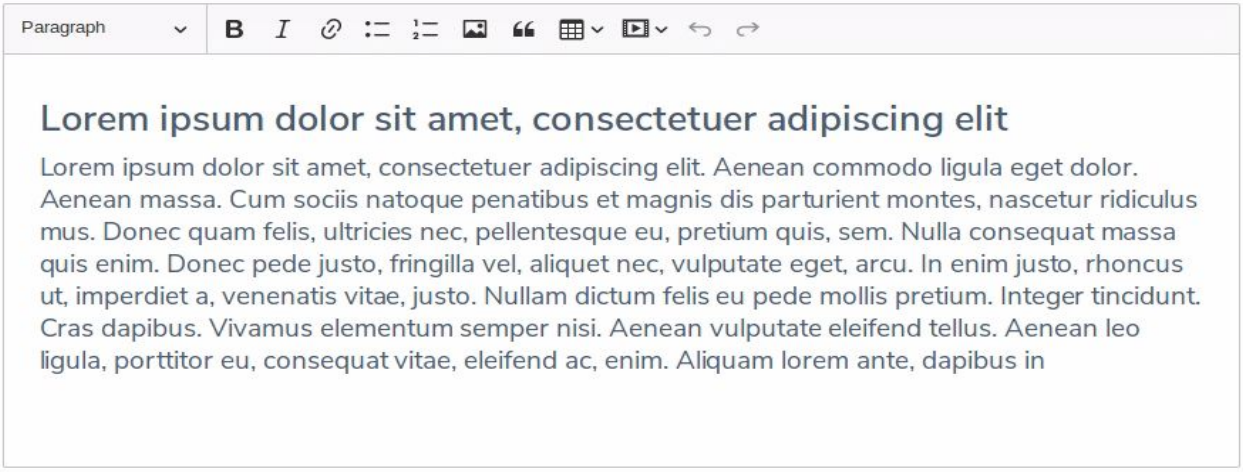
1.1 Why CKEditor 5 ?

CKEditor 5 Builds with new Design : CKEditor has introduced 4 new builds Classic, Ballon, Inline and Document and all these builds focus for different outcomes. By introducing new builds now CKEditor5 focus on each builds differently providing good developer experience

CKEditor 5 Framework : This will help us to build own text editor, this is completely customizable we can add/remove plugin based on needs this will help us in reducing space and load time of the creator dashboard. Further Ckeditor 5 has no plugin for KaTex, it has plugins only for MathJax. Ckeditor 5 Framework will help us in moving to KaTex in future and in building custom feature for the same.

1.2 Overview Ckeditor 5 Classic Build :

Ckeditor 5 Classic build is what oppia was traditionally used, it has few design changes like the editor grows with the size of content automatically, and now the editor is inline in the page without the surrounding <iframe> element this will help us in DOM Styling of the editor.



We can move to the **Classic Build** or to the **Ckeditor 5 Framework,** I'll take the classic build as the reference for this design doc. Further, i'd like to discuss more with the team on the selection of class build or framework

1.3 Approach to this migration :

Many things has been changed when we compare CKE4 with CKE5. The places we use CKEditor4 are Exploration_editor, Skill_editor, Story_editor and Topic_editor. First, the new CKEditor5 version needs to update in these files, after all the files are updated, the CkEditorRteDirective.js should be updated with new api calling used in CKEditor 5 Atlast pre plugin, Skins/ Theme and /extensions/rich_text_components should be regenerated with

CkEditorWidgetsInitializer.js. Manual Testing needs to be done in each phase of changes, Also the output data from CKEditor4 and Ckeditor 5 should be compared

1.3.1 Supporting changes needed for the files :

- **Changes in CKEditor Version :**

  Manifest.json
  scripts/install_third_party.sh

- **Changes in test files :**

  core/tests/karma.conf.js
  core/templates/dev/head/pages/tests/form_builder_test_page.html

- **Changes for Ckeditor 5 links and Design :**

  core/templates/dev/head/pages/story_editor/story_editor.html
  core/templates/dev/head/components/forms/schema_editors/
  Schema_based_html_editor_directive.html
  dev/head/pages/exploration_player/exploration_player.html
  core/templates/dev/head/pages/skill_editor/skill_editor.html
  core/templates/dev/head/pages/topic_editor/topic_editor.html
  dev/head/pages/exploration_editor/exploration_editor.html

- **Changes API Calling and Methods :**

  core/templates/dev/head/components/CkEditorWidgetsInitializer.js
  core/templates/dev/head/components/CkEditorRteDirective.js

- **Changes Ckeditor 5 plugins :**

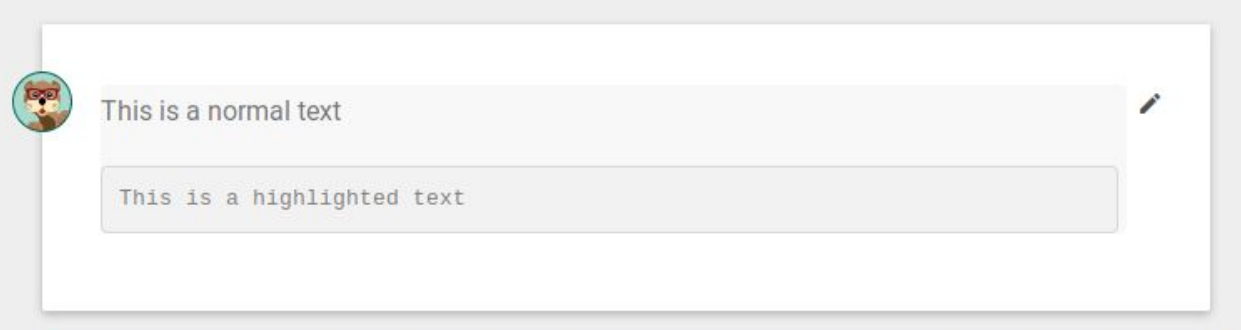
  extensions/ckeditor_plugins/pre/plugin.js
  core/tests/build_sources/extensions/plugin.js

- **Updates in oppia/extensions/rich_text_components/**

1.3.2 Supporting changes needed for extensions :

- **Regenerating Plugins for Ckeditor 5 :**

The implementation and methods used for creating a plugin in ckeditor are completely different and will require rewriting them from scratch. Functions and methods like CKEDITOR.plugins.add() editor.addCommand() are not not supported
The custom plugin we have now is
‘ pre ’ in **oppia/extensions/ckeditor_plugins/pre/plugin.js** the use of a pre plugin in CKEditor is to highlight a text in the exploration and other RTE Extensions in **/extensions/rich_text_components/** .



The pre plugin adds a <pre> </pre> to the texts in the current node whenever the pre icon is clicked in the CKEditor. In CKEditor4 the tag <pre> was identified using the CKEDITOR.style({element: 'pre'}); and the added to it using editor.addCommand().In Ckeditor 5 CKEDITOR.style() and editor.add are removed and now schemas are to built for the same.

CKEditor4	**CKEditor5**
CKEDITOR.plugins.add	Class extends plugin
CKEDITOR.style({element: 'pre'});	Editor.model.schema need to be created
editor.addCommand	editor.commands.add()
editor.ui.addButton()	view = new ButtonView(locale); view.set(button name);
editor.attachStyleStateChange()	view, 'execute', () => editor.execute(plug)

The regenerated code for the pre plugin is added below, similarly code for all the extensions in **/extensions/ckeditor_plugin/** should be regenerated

```
// Pre Plugin code for CKEditor 5

import Plugin from '@ckeditor/ckeditor5-core/src/plugin';
import ButtonView from '@ckeditor/ckeditor5-ui/src/button/buttonview';
import icon from 'icon/pre.png';
export default class PrePlugin extends Plugin {
init() {

const editor = this.editor;
const t = editor.t;

// Allow pre attribute on text nodes.
editor.model.schema.extend( '$text', { allowAttributes: pre } );
editor.model.schema.setAttributeProperties( pre, { isFormatting: true } );
// Build converter from model to view for data and editing pipelines.
editor.conversion.attributeToElement( {
          model: pre,
          view: 'pre',
    } );

editor.commands.add( pre, new AttributeCommand( editor, pre ) );
// Add pre button to feature components.
editor.ui.componentFactory.add( pre, locale => {
          const command = editor.commands.get( pre );
          const view = new ButtonView( locale );

          view.set( {
                label: t( 'pre' ),
                icon: icon,
                tooltip: true
          } );

view.bind( 'isOn', 'isEnabled' ).to( command, 'value', 'isEnabled' );
// Execute command.
this.listenTo( view, 'execute', () => editor.execute( pre ) );
return view;
  } );  } }
```

- **Regenerating Skins / Themes :**

*The **@ckeditor/ckeditor5-theme-lark** package contains the default theme of CKEditor 5, Further this theme can be edit with native styling. The previous theme/skin of CKE4 is not supported*

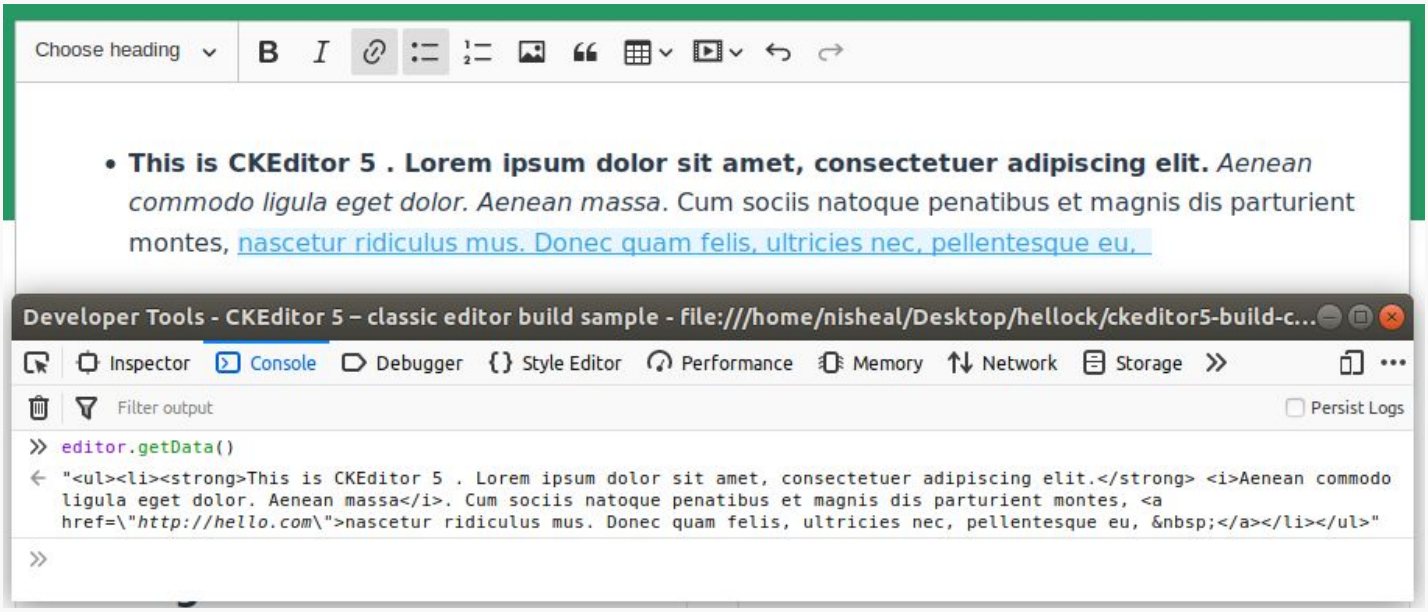
1.3.3 Check for Output-Data variation :

The Data produced by the CKEditor4 and our existing data may not be compatible with Ckeditor 5, there is no as such mentions about data change or incompatibility but the same need to be tested after the installation. If any data loss is observed the existing data need to be altered for Ckeditor 5 and jobs need to be created for the same.

To observe the output-data produced by both the editors I've installed CKE4 and CKE5 with the same input-data and the output-data can be retrieved using the below methods

***Method to check saved data in CKEditor4** :*
`CKEDITOR.instances.editor1.getData();`

Method to check saved data in CKEditor5 :
`editor.getData()`



CKEditor4	CKEditor5
`"<ul><li><strong>DATA 1</strong> <em>DATA 2</em>. DATA 3, <a href=\"url\">DATA 4</a></li></ul>"`	`"<ul><li><strong>DATA 1</strong> <i>DATA 2</i>. DATA3, <a href=\"url\">DATA 4 ,  </a></li></ul>"`
CKEDITOR.config.coreStyles_bold *{ element: 'b', overrides: 'strong' };*	*{ element: 'b', overrides: 'strong' };*
CKEDITOR.config.coreStyles_italic *{ element: 'i', overrides: 'em' }*	*{ element: 'i', overrides: '<i>' }*
CKEDITOR.config.coreStyles_underline = { *element: 'span',* *attributes: { 'class': 'Underline' }* *};*	*CKEditor 5 uses the <u> element* *<u> DATA 5 </u>*
CKEDITOR.config.coreStyles_strike = { element: 'span', attributes: { 'class': 'Strikethrough' }, overrides: 'strike'	*CKEditor 5 uses the <s> element*

};	
config.format_h1 = { element: 'h1', attributes: { 'class': 'contentTitle1' } };	*const headingConfig = {* *options: [* *{ model: 'paragraph', title: 'Paragraph', class: 'ck-heading_paragraph' },* *]* *};* *Similar with h2 : ck-heading_paragraph2*
no longer encapsulates the editing area in an <iframe>	*editing area can be easily controlled with CSS like :* *ck-content { width:400px; }*

The Comparison of Data produced by the CKEditor4 & CKEditor4 are different, jobs are need to migrate the existing data. In the job we will rewrite the existing data with the needed changes using beautifulsoup

***Example** : if we have a text “”*i love *oppia ”*

- *this text will be saved in CKE4 as [i love oppia]*
- *this text will be saved in CKE5 as [i love oppia]*

You can clearly see that the texts saved in both the cases are different, the previous CKE4 data will not be readable by the new CKE5.

a python-beautifulsoup scripts should be written to update all such existing data of CKE4

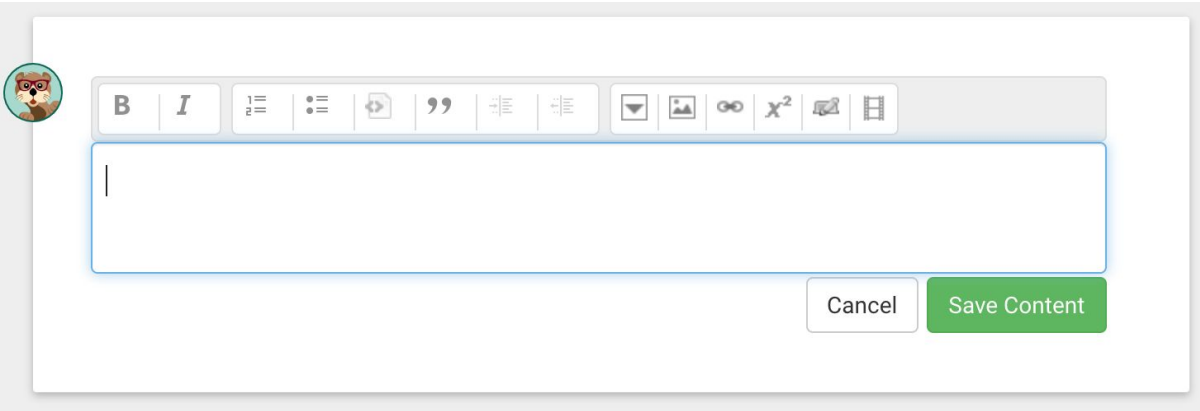
1.3.4 Tests needed for the Migration :

- *E2e Tests*
- *Backend tests to check for breaking changes*
- *Frontend tests*
- *Manual Testing for the data, plugins, extension and skin.*
- *Automated testing needed for data migration.*

1.2 Migration of CKE4 Plugins :

Earlier we had migrated to CKEditor 4 from the text-angular, In text-angular we had built **rich text components (RTC)** for image, Maths, link, tab and video which were converted to plugin/widgets for CKEditor-4 using apis hence, currently we use our own built plugin for CKE4 which is nothing but a RTC directive initialized as plugin/widgets.

Currently the plugin we have in our CKEditor 4 are :
We have few plugin which are default in CKEditor 4 , we have



- Default plugins - Plugins built by us - RTC Converted to plugins

.	Name	Use	Link
1.	Bold	To make text Bold	-
2.	Italic	To make text Italics	-
3.	Number List	For Numbered Lists	-
4.	Bulleted List	For bulleted Lists	-
5.	Pre	To Highlight a text in exploration	/oppia/ckeditor_plugins/pre
6.	Block-quote	To Insert Block-quote	-
7.	Increase Indent	To increase Indent	-
8.	Decrease Indent	To Decrease Indent	-
10.	collapse	To insert a Collapsible block	/oppia/extensions/rich_text_components/Collapsible

11.	Image	To insert images	/oppia/extensions/rich_text_components/Collapsible
12.	Link	To insert Links	/oppia/extensions/rich_text_components/link
13.	Maths	To insert Math Expressions	/oppia/extensions/rich_text_components/maths
14.	Tabs	To Insert tabs	/oppia/extensions/rich_text_components/tab
15.	Video	To insert Video content	/oppia/extensions/rich_text_components/video

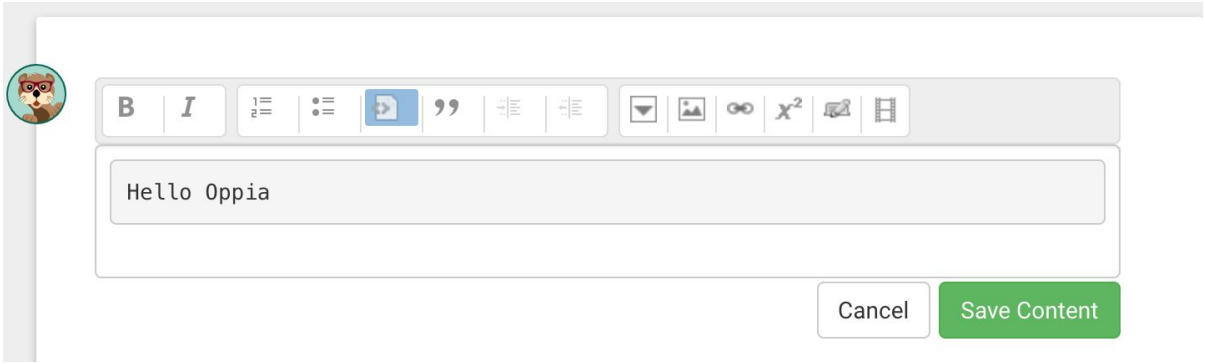
- **Default plugins** :

These are default and are directly added into the toolbar are easy to migrate

- **Plugins built by us** :

There is only one plugin built by us completely using the apis and it is used to highlight text in an exploration.

The previous apis which we used are all depreciated
in CKE hence, we need to write the plugin code from the scratch In CKEditor4 the tag <pre> was identified using the CKEDITOR.style({element: 'pre'}); and then added to editor using editor.addCommand().In Ckeditor 5 CKEDITOR.style() and editor.add are removed and now schemas are to built for the same.



 **- RTC Converted to plugins:**

Earlier we had migrated to CKEditor 4 from the text-angular, In text-angular we had built **rich text components (RTC)** like image, Maths, link, tab and video which were converted to plugin for CKEditor-4 using apis hence, currently we use our own built plugin for CKE4 which is nothing but a RTC directive.

We have created many plugin using this methods because it was convenient for us as we already had the RTC directives of text-angular. All the plugin their directive can be found in /oppia/extensions/rich_text_components/*

Now, create these RTC components into the CKE4 plugin we've used the same above mentioned apis

Here is the CkEditorWidgetsInitializer.ts which initialize all the RTC components as a CKE plugin hence, foreach RTC component a CKEDITOR.plugins.add() is initialized.

1.3 Challenges with CKEditor 5 :

Old Browser Compatibility Removed : CKEditor is not compatible with IE11 whereas, it provides full support to the latest stable version of many modern browsers.

Removed MathJax Plugin for CKE5 : CKEditor does not support MathJax Plugin issue here. An alternative **ckeditor5-math-preview** plugin must be added.

The **official Ckeditor 5 Maths Plugin** is expected to release in May updates here

2. ui-Sortable & jQuery Upgrade :

Enable a group of DOM elements to be sortable. Click on and drag an element to a new spot within the list, and the other items will adjust to fit. Oppia uses UI sortable in many pages like :

- **State_hints_editor_directive.html :** *To Drag and sort the hints*
- **State_responses_directive.html :** *To Drag and sort the state responses*
- **LearnerDashboard.html :** *To Drag and sort the play later exploration list*
- **Skill_concept_card_editor_directive.html :** *To Drag and sort the skills*
- **Answer_group_editor_directive.html**

2.1 Objective of this upgrade :

- Bugfix in ui-sortable will help us to solve issues in issue tracker.
- Ui-sortable compatibility with modern browsers
- Upgrading jQuery to provide smooth dragging behaviour for ui-sortable

2.2 Support Code changes needed :

Updating jquery -v to 3.3.1

- assets/scripts/embedding_tests_dev_0.0.1.min.html
- assets/scripts/embedding_tests_dev_0.0.2.min.html
- assets/scripts/embedding_tests_dev_i18n_0.0.1.html
- core/templates/dev/head/pages/header_js_libs.html

Updating jQuery t o 3.3.1 and ui-sortable to 0.19.0

- core/tests/karma.conf.js
- Manifest.json

2.2 Tests needed for the upgrade :

- *E2e Tests*
- *Backend tests to check for breaking changes*
- *Frontend tests*

- *Manual Testing for drag & drop behaviour.*
- *Manual Testing to check the upgraded jQuery version*

In each page.

After the successful upgrade, the applied css in the ui-sortable will be rechecked for improper position and overflow. Ui-sortable may provide unexpected results due to improper css.

***ui-sortable** and **jQuery** has been **successfully upgraded** in my forked repository [here]*

3. MathJax to 2.7.0 or KaTex :

A JavaScript display engine for mathematics, oppia useses MathJax in Exploration editor/player, Skill editor, Topic editor and Creator dashboard.

3.1 Comparison MathJax & KaTex :

When compared to MathJax , KaTex is very light weight an renders much faster. Whereas KaTeX supports a limited subset of the functionality provided by MathJax.

```
\frac{1}{\Bigl(\sqrt{\phi \sqrt{5}}-\phi\Bigr) e^{\frac25 \pi}}
\equiv 1+\frac{e^{-2\pi}} {1+\frac{e^{-4\pi}} {1+\frac{e^{-6\pi}}
{1+\frac{e^{-8\pi}} {1+\cdots} } } }
```

The above Latex is processed by **MathJax in 2097 ms** and by **KaTex in 257 ms** the live comparison can be done here

3.2 Challenges will KaTex :

We use CKeditor as the only means to input Mathematical expression and CKEditor 5 has no support for plugins for KaTex and MathJax. Third-party plugin provide good stable support for MathJax like ***Ckeditor5-math-preview,*** *but the support for KaTex is still a question*

The ***objective*** *of this upgrade will be to update MathJax and bring codebase in a stage where migration to KaTex can be carried out easily in future, the implementation of* ***Ckeditor5-math-preview*** *plugin with provide us an option to switch between the render engine*

3.3 Support Code changes needed :

Updating MathJax to -v to 2.7.0

- Manifest.json
- scripts/install_third_party.sh
- scripts/install_third_party.py

Updating script path to -v to 2.7.0

- dev/head/pages/story_editor/story_editor.html
- dev/head/pages/exploration_player/exploration_player.html
- dev/head/pages/skill_editor/skill_editor.html
- dev/head/pages/topic_editor/topic_editor.html
- dev/head/pages/exploration_editor/exploration_editor.html
- dev/head/pages/creator_dashboard/creator_dashboard.html

Upgrading MathJax to the latest version is easier as no breaking changes have been observed while upgrading to the latest version. But CKEditor 5 has depreciated the MathJax plugin from the CKEditor 5 and now there is no official plugin for MathJax.

We have to make use of the third party plugin ***ckeditor5-math-preview*** *stable version tested with CKE5 v11.2.0.*

Ckeditor5-math-preview *has the render engine for both MathJax and KaTex so this plugin will help us to move into KaTex in future as it is easy to change the render engine.*

3.4 Implementation of Ckeditor5-math-preview plugin :

- To Import **Ckeditor5-math-preview** plugin :

```
import MathpreviewPlugin from 'ckeditor5-math-preview/src/mathpreview'
```

- Configuring the editor :

```
ClassicEditor.builtinPlugins = [
MathpreviewPlugin,
 ],

ClassicEditor.defaultConfig = {
        toolbar: {
             items: [
                        'mathpreview',
                  ]
        },
   },

MathPreview = {
        //configuration options for the math-preview plugin
        engine: 'mathjax' //or 'katex'
   },
```

3.4 Testing the plugin and MathJax Upgrade :

- .Backend tests to check for breaking changes
- Manual Testing of the plugin .
- Checking the render Speed with complex latex
- Manual testing for page load time

4. Upgrading GAE (gae.pipeline, google-sdk, gaeCloudStorage)

GAE Libraries are very important to us, the below are the important GAE libraries and their latest versions

gaeCloudStorage	1.9.15.0	1.9.22.1
gaeMapReduce	1.9.17.0	1.9.84
gaePipeline	1.9.17.0	1.9.84
google-sdk	221	242

- ***4.1 gaeCloudStorage*** : *This library helps in accessing Google Cloud Storage from App Engine, To upgrade this library common support changes are needed. The latest library version should be updated to* 1.9.22.1 *in all the needed pages*

- **4.2 gaeMapReduce** : *This library helps in processing large amounts of data in a parallel and distributed. To upgrade this library common support changes are needed. The latest library version should be updated to* 1.9.84 *in all the needed pages*

- **4.3 gaePipeline** : *This library helps in connecting together various App Engine MapReduce into a computational pipeline. To upgrade this library common support changes are needed. The latest library version should be updated to* 1.9.84 *in all the needed pages*

- **4.4 Google-sdk** : *Google Cloud SDK manage resources and applications hosted on Google Cloud Platform. To upgrade this library common support changes are needed. The latest library version should be updated to* 242 *in all the needed pages*

Tests Needed :

- ***Common Tests*
- *Manually test the datastore.*
- *Running all the jobs to check for datastore interaction.*
- *Running all jobs to test MapReduce*
- *Building and Running the application in different platforms*

5. Migrating to Font-Awesome 5 :

3 Different Icon Styles - We introduced a solid, regular, and light style of every icon in Font Awesome. Fa is no longer a valid class in Font-Awesome and now Font-Awesome has separated free and premium icons using fal and far / fas classes

Font-Awesome 4 syntax	***Font-Awesome 5 syntax***
**	**

Outline Style Icons *- all icons that had an outlined style (and usually ended with -o) now have a prefix of far and have had their -o suffix removed.*

We use -o for exploration rating stars in ***RatingDisplayDirective.js*** *and its test in* ***RatingDisplayDirectiveSpec.js*** *which should be changed.*

5.1 Supporting Changes needed :

5.1.1 Updating new Class Names :

*After the **common changes the new classes name should be updated and fa should be removed from the codebase. We use font-awesome in many places including exploration and learner dashboard all the files should be changed with the latest coding style*

Comparing the classnames (' font-awesome 4 ',' font-awesome 5 ') :

('fa fa-star-half-empty', 'far fa-star-half'), *('fa fa-star-half-full', 'far fa-star-half'),* *('fa fa-star-half-o', 'far fa-star-half'),* *('fa fa-star-o', 'far fa-star'),* *('fa fa-star', 'far fa-star'),* *('fa fa-arrows-alt', 'fas fa-expand-arrows-alt'),* *('fa fa-arrows-h', 'fas fa-arrows-alt-h'),* *('fa fa-arrows-v', 'fas fa-arrows-alt-v')*	*('fa fa-arrow-circle-o-down', 'far fa-arrow-alt-circle-down'),* *('fa fa-arrow-circle-o-left', 'far fa-arrow-alt-circle-left'),* *('fa fa-arrow-circle-o-right', 'far fa-arrow-alt-circle-right'),* *('fa fa-arrow-circle-o-up', 'far fa-arrow-alt-circle-up'),* *('fa fa-arrows', 'fas fa-arrows-alt'),*

For all the changed classnames follow here

Script/Job to update the new class-name (far, fas) :

```
import os
from collections import OrderedDict

findreplace = [ ('fa facebook',’fas fa facebook’),
    ('fa fa-arrow-circle-o-left', 'far fa-arrow-alt-circle-left'),
    ('fa fa-star', 'far fa-star'),
    ('fa fa-times', 'far fa-clock'),]
#MORE NEW CLASSES SHOULD BE ADDED
def upgrade(project_path, extensions, exclude_directories):
    for dname, dirs, files in os.walk(project_path):
       dirs[:] = [d for d in dirs if d not in exclude_directories]
       for fname in files:
            if(fname.lower().endswith(extensions)):
                 fpath = os.path.join(dname, fname)
                 print fpath
                 s = open(fpath).read()
                 for icon in findreplace:
                      s = s.replace(icon[0]+'"', icon[1]+'"')
                      s = s.replace(icon[0]+"'", icon[1]+"'")
                 f = open(fpath, "w")
                 f.write(s)

print "\nAbove files have been modified"
project_path = os.getcwd()

extensions = ('.html', 'js','css')
exclude_directories = ‘’
upgrade(project_path, extensions, exclude_directories)
```

New Updated icons :

Font-awesome 4	***Font-awesome 5***
	

5.2 Challenges while upgrading :

4.2.1 Changed Directory and paths in Font-Awesome 5

Folder	File
/css	All.css
/js	All.css
/webfonts	Contains all webfonts

Font-Awesome 5 has all the fonts in `fonts` folder hence it was easy to combine FA4 into the third_party/generated/third_party.css. In FA5 the font folder has been renamed to webfonts, it FA5.css is combined to third_party/generated/third_party.css a folder call is made to ../webfont which is not available in third_party/generated/.
third_party/generated/ has only common `font` folder for use, hence FA5.css cannot be combined in third_party/generated/third_party.css (suggestion need).
For now i've added a link in header_css_lib.css for FA5.css

5.2.2 Updating front-end test :

Frontend test files should be updated for all the classes and ***RatingDisplayDirectiveSpec.js*** *should be re-checked and updated with the new class names implemented in* ***RatingDisplayDirective.js****.*

5.3 Tests Needed :

- ***common tests.*
- *Check the footers for social media icons.*
- *Test the Exploration ratings for star icon (Rating an Exploration).*
- *Manual Testing for Audio play pause icons*
- *Check all the pages where the fa-icons are updated to see if needed icons are shown up and are relatable to its content.*

*I've **raised a PR** for Font-Awesome Migration [here]*

6. Migrating to Bleach 3 with html5lib *:*

Bleach is an allowed-list-based HTML sanitizing library that escapes or strips markup and attributes. It helps in for sanitizing text from untrusted sources.

Oppia uses bleach for validate the content and sanitize HTML.

6.1.1 Support Changes needed :

Oppia widely uses bleach.clean() which is now rewritten, bleach needs 3 extra libraries callable, Webcoding and Six to support bleach.clean() and now callables can call three argument.

We use bleach.clean() in html_cleaner.py and filter_a function should be written for 3 parameters ***filter_a(tag, arrt, val)***

- ***Common Changes Needed*
- *More parameter to filter_a of html_cleaner.py*
- *Updating backend tests*

The backend test written in exp_domain_test.py , param_domain_test.py needs an update. Bleach cannot take dict input they need to be converted into the text input hence, '{}' should be replaced with ''

6.1.2 Tests needed :

- ***Common Tests*
- *Manual testing of data validation*
- *Observing the return data of bleach html cleaner*

I've ***successfully upgraded bleach*** *in my forked repo here.*

7. Upgrading bower based Angular static libraries

Oppia widely uses bower based angular libraries in translation and cookie.

The identified libraries are :

Library	***W.r.t bower***	***Upgrading to***
bowerMaterial 0.6.0-rc1		*1.1.18*
angularTranslate 2.8.1	*bower-angular-translate*	*2.18.1*
angularTranslateLoader 2.8.1	*bower-angular-translate-loader-static-files*	*2.18.1*
angularCookiesRev 1.4.0	*bower-angular-cookies*	*1.6.2*
angularStorageCookiesRev 2.8.1	*bower-angular-translate-storage-cookie*	*2.18.1*
angularTranslateInterpolationMessageFormat 2.11.0	*bower-angular-translate-interpolation-messageformat*	*2.18.1*
bowerAngularTranslateLoaderPartial 2.7.1	*bowerAngularTranslateLoaderPartial*	*2.18.1*

7.1 Supporting changes needed

- ***Bower-angular-cookies :*** *The latest version for bower-angular-cookies is 1.7.8 but the version after 1.6.2 uses module.info function which need angular > 1.6.2*
- ***** Common Changes*** *: Other bower based libraries needs only common changes they don't have any breaking changes and supporting changes*

7.2 Tests Needed :

*I've **successfully upgraded bowerMaterial** in my repo here.*

*And other bower based libraries **angularTranslate**, **angularCookiesRev**, **angularStorageCookiesRev**, **angularTranslateInterpolationMessageFormat**, **bowerAngularTranslateLoaderPartial**, here*

8. Migrating to BootStrap 4

The BootStrap 4 is a complete rewritten of BootStrap 4, there is no breaking changes but many of the class names are depreciated or renamed.

Switched from px to rem as our primary CSS unit, though pixels are still used for media queries and grid behavior as device viewports are not affected by type size

8.1 Classes Renamed :

	Bootstrap 3	***Bootstrap 4***
Typography	*.page-header* *.dl-horizontal* *<blockquote>*	*Dropped* *dropped* *.blockquote*
Buttons	*.btn-default* *.btn-xs*	*.btn-secondary* *dropped*
Dropdowns	*.divider*	*.dropdown-divider*
Image	*.img-responsive* *.img-circle* *.img-rounded*	*.img-fluid* *.rounded* *.rounded-circle*
Navbar & Navs	*.navbar-default* *.navbar-toggle* *.navbar-form* *.navbar-btn* *.navbar-right* *.navbar-left* *.navbar-fixed-top* *.nav > li* *.nav > li > a*	*.navbar-expand-{breakpoint}* *.navbar-light* *.navbar-toggler* *.form-inline* *.nav-item* *.ml-auto* *.mr-auto* *.fixed-top* *.nav > li.nav-item* *.nav > li.nav-item > a.nav-link*

Panel	*.panel* *.panel-default* *.panel-group* *.panel-heading* *.panel-title* *.panel-body* *.panel-footer*	*.card* *dropped* *.card-group* *.card-header* *.card-title* *.card-body* *.card-footer*

8.2 *Supporting Changes needed :*

Many of the above mentioned classes of bootstrap 3 are used by oppia jobs/scripts should be written to renamed these classes

- ***Common Changs need*
- *All the BootStrap 3 class names should be renamed to BootStrap 4*
- *Nav-default is widely used in codebase which should be changed to navbar-light*
- *Respective class names should be updated in the frontend test*

Script/Job to update the new class-name :

```
import os
from collections import OrderedDict

findreplace = [ ('navbar-default',’navbar-light’),
    ('img-circle', 'rounded')
]
#MORE NEW CLASSES SHOULD BE ADDED
def upgrade(project_path, extensions, exclude_directories):
    for dname, dirs, files in os.walk(project_path):
       dirs[:] = [d for d in dirs if d not in exclude_directories]
       for fname in files:
            if(fname.lower().endswith(extensions)):
                 fpath = os.path.join(dname, fname)
                 print fpath
                 s = open(fpath).read()
                 for icon in findreplace:
                      s = s.replace(icon[0]+'"', icon[1]+'"')
                      s = s.replace(icon[0]+"'", icon[1]+"'")
                 f = open(fpath, "w")
                 f.write(s)

print "\nAbove files have been modified"
project_path = os.getcwd()

extensions = ('.html', 'js','css')
exclude_directories = ‘’
upgrade(project_path, extensions, exclude_directories)
```

8.3 *Tests needed :*

- **Common Tests
- Check if all the classes are renamed classes using the grep commands
- Test the Navbar and Exploration cards in all screen sizes.

Manual Testing of pages in mobile view

Classification of ./third_pary libraries based on analysis :

**The libraries which are already in their latest version and angular statics are excluded. i'v e included them in my timeline*

1.  **Salient Upgrades :**

Upgrading these libraries will provide significant feature improvements to the codebase and needs supporting code changes for Implementation & Testing.

2.  **Moderate Upgrades :**

Upgrading these libraries will support the existing infrastructure in bug-fix and needs supporting code changes for Implementation & Testing.

3.  **Easy Upgrades :**

These libraries are easy to upgrade and upgrading these libraries will support the codebase indirectly.

	Library	Current Version	Latest Version
1.	jquery	3.2.1	3.3.1
2.	uiSortable	0.17.1	0.19.0
3.	fontAwesome	4.7.0	5.8.1
4.	bootstrap	3.3.4	4.3
5.	Bleach	1.2.2	3.1.0
6.	CKEditor	4	5

	Library	Current Version	Latest Version
1.	waveSurferJs	1.0.46	2.2.1
2.	gaeMapReduce	1.9.17.0	-
3.	gaePipeline	1.9.17.0	-
4.	html5Lib	0.95	1.0.1
5.	requests	2.10.0	2.21.0
6.	angularTranslateInterpolationMes	2.11.0	2.18.0

	sageFormat		
7.	mutagen	1.38.0	1.43.0
8.	messageFormat	0.3.1	2.0.5
9.	mathJs	4.2.2	5.8.0
10.	midiJs	-	0.3.0
11.	hammerJs	2.0.4	2.0.8
12.	guppy	-	2.0.0
13.	select2	4.0.3	4.0.6
14.	ckeditorSharedspace	4.9.2	4.11.3
15.	uiCodemirror	0.1.2	0.3.0
16.	codemirror	3.19.0	5.45.0

	Library	Current Version	Latest Version
1.	gaeCloudStorage	1.9.15.0	1.9.22.1
2.	angularAudio	-	1.7.3
3.	angularTranslate	2.8.1	2.18.1
4.	angularCookiesRev	1.4.0	1.7.8
5.	angularStorageCookiesRev	2.8.1	2.18.1
6.	simpleJson	3.7.1	3.16.1
7.	mathExpressions	1.7.0	2.0.0.alpha
8.	angularToastr	1.7.0	2.1.1
9.	soupsieve	1.8	1.9.0
10.	ngJoyride	5b25d076405c9a787db602d2a8ff49919228f4fc	0.1.20
11.	uiUtils	0.1.1	-
12.	browsermob-proxy	2.1.1	2.4.1
13.	ngImageCrop	0.3.2	-

Upgrading Node to the latest stable version:

- *Current Node -v :* **Node-v6.9.1**
- *Latest Stable Node -v :* **Node-v10.15.3**
- **Support changes needed :**
  - **/scripts/setup.sh**

    Changing Node reference path & filename to latest version :

```
export NODE_PATH=$TOOLS_DIR/node-10.15.3
```

    Updating the curl command for Node's latest -v file path

```
curl -o node-download.tgz
https://nodejs.org/dist/v10.15.3/$NODE_FILE_NAME.tar.gz
```

  - **/scripts/build.py**

    Updating Node directory to `node-10.15.3`

  - **/scripts/pre_commit_linter.py**

    Updating Node directory to `node-10.15.3` in **_lint_html_files** & **_pre_commit_linter**

Classification of Libraries in ./node_modules :

Salient Upgrades :

Moderate Upgrades :

Easy Upgrades :

Deprecated :

	Library	Current Version	Latest Version
1.	ajv	5.0.0	6.10.0
2.	babel-eslint	10.0.1	10.0.1
3.	browserstack-local	1.3.3	1.3.7
4.	dotenv	6.0.0	7.0.0
5.	eslint	4.19.0	5.16.0
6.	eslint-plugin-angular	0.12.0	4.0.0
7.	eslint-plugin-html	4.0.1	5.0.3
8.	gulp	3.9.0	4.0.0
9.	gulp-clean-css	2.0.2	4.0.0
10.	gulp-concat	2.6.0	2.6.1
11.	gulp-sourcemaps	1.6.0	2.6.5
12.	gulp-uglify	2.0.1	3.0.2
13.	gulp-util	3.0.7	Deprecated, Easy Migration to vinyl
14.	htmllint	0.7.2	0.7.3
15.	htmllint-cli	0.0.7	0.0.7
16.	@mapbox/stylelint-processor-arbitrary-tags	0.2.0	0.2.0
17.	postcss-syntax	0.10.0	0.36.2
18.	stylelint	9.2.1	9.10.1

19.	stylelint-config-standard	18.2.0	18.2.0
20.	through2	2.0.0	3.0.1
21.	typescript	3.3.3	3.4.2
22.	@types/angular	1.6.54	1.7.0
23.	@types/angular-animate	1.5.10	1.5.10
24.	@types/angular-mocks	1.7.0	1.7.0
25.	@types/ckeditor	4.9.2	4.9.8
26	@types/d3	3.5.40	5.7.2
27	@types/google.visualization	0.0.46	0.0.46
28	@types/jasmine	3.3.8	3.3.12
29	@types/jasmine-jquery	1.5.33	1.5.33
30	@types/jasminewd2	2.0.6	2.0.6
31	@types/jquery	3.3.29	3.3.29
32	@types/jqueryui	1.12.1	1.12.7
33	@types/leaflet	1.4.0	1.4.3
34	@types/mathjax	0.0.35	0.0.35
35	@types/mathjs	5.0.0	5.0.1
36	@types/mousetrap	1.6.1	1.6.2
37	@types/node	6.14.3	11.13.0
38	@types/select2	4.0.48	4.0.48
39	@types/q	1.5.1	1.5.2
40	@types/selenium-webdriver	2.53.43	4.0.0
41	uglify-js	3.3.11	3.5.3
42	yargs	3.29.0	13.2.2
43	jasmine-core	2.5.2	3.4.0
44	karma	1.5.0	4.0.1
45	karma-jasmine	1.1.0	2.0.1
46	karma-jasmine-jquery	0.1.1	0.1.1
48	karma-coverage	1.1.1	1.1.2

49	karma-ng-html2js-preprocessor	1.0.0	1.0.0
50	karma-chrome-launcher	2.0.0	2.2.0
51	protractor	5.3.1	5.4.2
53	jasmine-spec-reporter	3.2.0	4.2.`0
54.	pegjs	0.8.0	0.10.0

Analysis of Libraries in ./oppia_tools :

Library	**Changelog & Support Changes**	**Remarks**
Pylint **(** 1.9.3 **to** 2.4.0 **)**	*Changelog :* • *Updates to logging-format-style*	Moderate to upgrade
Pylint-quotes **(** 0.1.9 **to** 0.2.1 **)**	*Changelog :* • *Bug Fix*	Easy to upgrade
Webtest **(** 1.4.2 **to** 2.0.34 **)**	*Changelog :* • py33 is no longer supported *Support Changes : None*	Easy to upgrade no support code changes needed
Isort **(** 4.2.15 **to** 4.3.16 **)**	*Changelog :* • Python 3.x support *Tests Needed :* • *Manual testing with import statements* • *Backend tests for breaking changes*	Isort is much need to validate the alphabetical sorting of the import statements. New version has no breaking changes moderate to upgrade

Pycodestyle **(** 2.3.1 **to** 2.5.0 **)**	*Changelog :* ● *--max-doc-length* ● *support for Py 3.x*	Easy to upgrade
PyGithub **(** 1.43.5 **to** 1.43.6**)**	*Changelog :* ● *Remove Status API*	Easy to upgrade
Google-cloud-sdk **(** 222.0.0 **to** 241.0.0 **)**	*Changelog :* ● *gcloud datastore command updated* ● *Upgrades in installation process* *Support Changes :* ● *gcloud datastore should be updates with latest installation command* *Tests Needed :* ● *Manual Testing needed with datastore*	DataStore and sdk installation commands are changed, the same needs to be updated in the gaesuite Moderate to upgrade
Coverage **(** 4.5.1 **to** 4.5.3 **)**	*Changelog :* ● *Python 3.x Support* ● *Bug Fix* *Support Changes :* ● *Only Version Changes*	Moderate to upgrade

Classification of ./oppia_tools based on analysis:

Salient Upgrades :

Moderate Upgrades :

Easy Upgrades :

Easy Upgrades :

	Library	Current version	Latest Version
1.	skulpt	0.10.0	-
2.	pylint	1.9.3	2.3.1
3.	pylint-quotes	0.1.9	0.2.1
4.	webtest	1.4.2	2.0.34
5.	isort	4.2.15	4.3.6
6.	pycodestyle	2.3.1	2.5.0
7.	esprima	4.0.1	4.0.1
10.	PIL	1.1.7	1.1.7
11.	PyGithub	1.43.5	1.43.6
13.	google-cloud-sdk	222.0.0	241.0.0
14.	google_appengine	1.9.67	1.9.67
15.	esprima	4.0.1	4.0.1
16.	coverage	4.5.1	4.5.3

Milestones :

After the classification of all the libraries and modules
*In **./third_party** we have about 36 libraries, the distribution is below:*

Library	*Library Count*	*Total Upgrade bit*
Salient libraries *Upgrade bit : 8*	4	32
Moderate libraries *Upgrade bit : 5*	19	95
Easy Upgrades *Upgrade bit : 2*	13	26

= 153

***Milestone 1** : Upgrade 70% of the libraries in ./third_party, with no regressions.*

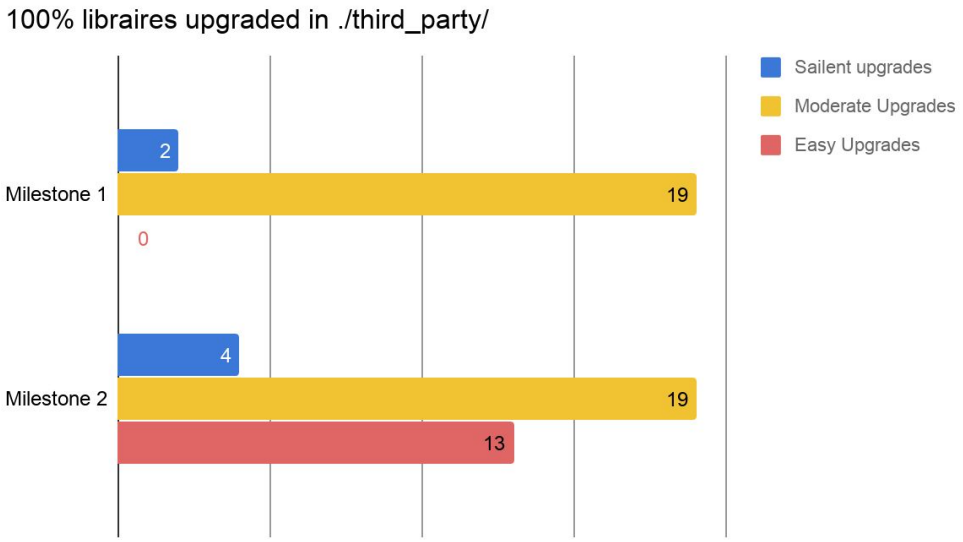
*In Milestone 1, I'll update **2 salient libraries** and **all the moderate libraries**.*

*The total upgraded bit in milestone is 111, % update is 111/153 = **73%***

***Milestone 2** : Upgrade 100% of the libraries in ./third_party, with no regressions.*

*In Milestone 1, I'll update **2 salient libraries** and **all the Easy upgrade libraries**.*

*The total upgraded bit in milestone is 42, % update is 42/153 = **27%***



*In **./node_modules** we have about 36 upgradable libraries,*

the distribution is below :

Library	***Library Count***
Salient libraries	5
Moderate libraries	22
Easy Upgrades	9

*In **./oppia_tools** we have about 36 upgradable libraries,*

the distribution is below :

Library	***Library Count***
Moderate libraries	7
Easy Upgrades	2

***Milestone 3** : Upgrade Node to the latest stable version and then upgrade all libraries in ../node_modules. Upgrade all libraries in ../oppia_tools.*

In Milestone 3, First Node will be upgraded to the latest stable version and the upgraded will be tested for the breaking changes.

After the successful upgrade of node, all the node modules will be upgraded in increasing order of their upgrade value with backend tests.

At last, libraries in /oppia_tools/ will be updated and tested in increasing order of their upgrade value.

Timeline :

May 27 to August 19 (12 Weeks) :

** 2 Days buffer after every milestone.					
Milestone	***Week***		***Library***	***PR Raised***	***PR Merged***
Milestone 1 *May 27 - 21 June*	*Week 1* *27 May - 02 June*	*1.1.1*	***Bleach***	*28 May*	*30 May*
		1.1.2	*waveSurferJs*	*29 May*	*31 May*
		1.1.3	***gaeMapReduce***	*30 May*	*01 June*
		1.1.4	***gaePipeline***	*31 May*	*02 June*
		1.1.5	*html5Lib*	*01 June*	*03 June*
		1.1.6	*requests*	*02 June*	*04 June*
	Week 2 *03 June - 09 June*	*1.2.1*	*angularTranslateInterpolation MessageFormat*	*04 June*	*06 June*
			mutagen		
		1.2.2	*messageFormat*	*05 June*	*07 June*
		1.2.3	*fontAwesome*	*06 June*	*08 June*
		1.2.4	*mathJs*	*07 June*	*09 June*
		1.2.5	*midiJs*	*08 June*	*10 June*
		1.2.6	*hammerJs*	*09 June*	*11 June*
	Week 3 *10 June - 16 June*	*1.3.1*	*uiCodemirror*	*11 June*	*12 June*
		1.3.2	*Codemirror*		
		1.3.3	*ckeditorSharedspace*	*12 June*	*14 June*
		1.3.4	*guppy*	*13 June*	*15 June*
		1.3.5	***Bootstrap***	*16 June*	*18 June*

	Week 4 *17 June - 21 June*	*1.4.1*	***Salient upgrade 1 :*** *upgrading ui-sortable and jquery, fixing ui-sortable issues.*	*20 June*	*21 June*
Milestone 2 *24 June - 19 July*	*Week 5 - Week 6* *24 June - 7 July*	*2.1.1*	***Salient upgrade 2 :*** *CKEditor 4 - 5 Migration*	*4 July*	*8 July*
	week 7 *8 July - 14 July*	*2.3.1*	***Salient upgrade 3 :*** *MAthJax upgrade and plugin integration*	*12 July*	*14 July*
	Week 8 *15 July - 19 July*	*2.4.1*	***All Angular static Libraries***	*16 July*	*17 July*
		2.4.2	*simpleJson* *mathExpressions*	*17 July*	*18 July*
		2.4.3	*soupsieve*	*18 July*	*19 July*
		2.4.4	*ngJoyride*	*19 July*	*20 July*
		2.4.5	*uiUtils*		
Milestone 3 *22 July - 16 Aug*	*Week 9* *22 July - 28 July*	*3.1.1*	***Upgrade Node to Latest Stable Version***	*27 July*	*29 July*
	Week 10 *29 July - 4 Aug*	*3.2.1*	*browserstack-local* *dotenv*	*30 July*	*31 July*
		3.2.2	***eslint*** *eslint-plugin-angular* *eslint-plugin-html*	*31 July*	*02 July*
		3.2.3	*gulp* *through2*	*01 July*	*03 July*
		3.2.4	*postcss-syntax*	*02 July*	*04 July*

		3.2.5	stylelint	03 July	05 July
			htmllint		
		3.2.6	typescript	04 July	06 July
	Week 11 05 Aug - 11 Aug	3.3.1	@types/angular ckeditor @types/ckeditor @types/mathjs	6 Aug	7 Aug
		3.3.2	@types/mousetrap @types/node @types/q	6 Aug	8 Aug
		3.3.3	@types/jasmine @types/leaflet @types/d3	6 Aug	8 Aug
		3.3.4	@types/selenium-webdriver	7 Aug	8 Aug
		3.3.5	uglify-js	7 Aug	8 Aug
		3.3.6	yargs	7 Aug	8 Aug
		3.3.7	jasmine-core	8 Aug	9 Aug
		3.3.8	karma	8 Aug	9 Aug
		3.3.9	karma-jasmine	9 Aug	10 Aug
		3.3.10	karma-coverage	9 Aug	10 Aug
		3.3.11	karma-chrome-launcher	10 Aug	12 Aug
		3.3.12	protractor	10 Aug	12 Aug
		3.3.13	jasmine-spec-reporter	10 Aug	12 Aug
		3.3.14	pegjs	11 Aug	12 Aug
	Week 12 12 Aug - 17 Aug	3.4.1	**google-cloud-sdk**	13 Aug	14 Aug
		3.4.2	PyGithub	14 Aug	16 Aug
			pycodesty		
			isort		
			webtestle		
		3.4.3	**pylint**	15 Aug	16 Aug
			pylint-quotes		
		3.4.4	esprima	16 Aug	17 Aug
			coverage		

Devlogs :

I've planned to maintain a google spreadsheet where the upgrade activity and the work progress can be tracked by my mentor and team members.

Summer plan :

I've no summer commitments, I'll be dedicating time in oppia.

Timezone :

I'll be in Indian Standard Time (IST) (UTC + 05:30) throughout the project timeline.

Time to commit for the project and other obligations :

I've my holidays from 10 May to 10 July, I can dedicate 7 hours a day for my project during weekends and holidays. During college, after 10th July I can dedicate 5 - 6 hours daily. This will be Extended if the project demands more.

.

Date	***Hours a Week***
10th May - 10th July	*49 - 50*
10 July - 19th Aug	*40 - 42*

Communication :

- ### Contact Information :

Nisheal A John

Email : *nishealajohn@karunya.edu.in*

Phone : *(+91) 8072137030*

Github : *@NishealJ*

Linkedin : */in/NishealJohn*

I'm always available on Hangout, Gitter and on emails, I can connect with my mentor in any one of the above platforms. This is open to suggestions if needed I'll be interested to connect in other platforms as well.