

# Google Summer of Code 2022

## Angular Migration

Shivam

### Section 1: About You

My name is Shivam. I am a second-year undergraduate student at the National Institute of Technology Hamirpur, pursuing Electrical Engineering as my major.

#### What project are you applying for?

Migration from angularJs to Angular11

#### Why are you interested in working with Oppia, and on your chosen project?

I started contributing to Oppia in October 2021. The reason I started contributing to Oppia was that I want to explore the open-source community and I really want to participate in GSoC. Contributing to Oppia helped me in learning Angular, AngularJS introduced me to the test-driven environment, and whenever I got stuck mentors were readily available for help.

I learned a lot about Angular, Frontend unit testing, and E2E testing while working with the migration team. Now, I am quite familiar with the source code. I worked on the migration from day one and have a good understanding of the issues that can arise and how to address them; finishing this project would be a great learning experience for me and I also want to remove the hybrid mode i.e. (Angular & AngularJs) from the project to make this more developer-friendly.

#### Prior Experience

I've been doing web development for the past one and a half years, and I'm a front-end web developer who predominantly uses the Angular/React frontend frameworks. For the past 5-6 months, I've been contributing to Oppia as part of the angular migration team.

1. I am a member of the angular migration team in Oppia. I have been regularly contributing from Oct 2021 to the migration project. My contributions involve creating PRs, helping other team members, creating bug issues present in the project, and also resolving them. At present I have 24 migration PRs merged and 5 issues closed.
2. I am also a member of the oppia web welfare team where we help other contributors, till now I am successfully able to answer 1 contributor query and tried my best to address another query too.
3. I have also developed a [website](#) for The Robotics Society for an event called Roboweeek built with React.
4. I have also made a [portfolio website](#) built on angular.

### Links to previous PRs.

- [#14798](#) Customize-interaction-modal from AngularJS to Angular.
- [#14873](#) State-interaction-editor from AngularJS to Angular.
- [#14576](#) State-hints-editor, response-header from angularJS to Angular.
- [#14509](#) Resolved Issue [#14502](#) Reusing save pending changes modal.
- [#14505](#) Resolved Issue [#14501](#) + AngularJs to Angular Migration.

A complete list of my PRs can be found [here](#).

I have opened 5 issues and fixed them which can be found [here](#).

I did volunteer work in resolving the E2E Test and FrontEnd Test in [Srijan Reddy Vasa Schema-based-editor PR](#), my PRs can be found [here](#).

### Project Size

I am applying for a large project (~350 hours).

## Project Timeframe

13 June 2022 to 3 Oct 2022

## Contact info and timezone(s)

**Primary Email and Hangout:** [shivamthalwal@gmail.com](mailto:shivamthalwal@gmail.com)

**Secondary Email:** [20bee095@nith.ac.in](mailto:20bee095@nith.ac.in)

**Contact Number:** +91 8219601395

**GitHub:** @heyimShivam

**Discord:** heyimshivam#4662

**Time Zone:** Kolkata, India (GMT+5:30)

**Preferred mode of Communication:** Hangout, Email, Google meet, Discord.

## Time Commitment

I am having summer vacation from 1 June to 16th July. In this time frame, I would be able to devote **~42 hr/week**. After that, I would be able to devote **~21.5 hr/week**. which may increase if the need arises. I am working on the GSoC project from 13th June to 3rd Oct timeframe ( **Note:** can be extended if the need arises).

S. No	Dates	Days (Total)	Time Commitment
1.	13th June - 20th July	Mon-Sat (6)	7 hr/day (Mon-Sat)
2.	20th July - 12th Sep	Mon-Sat (6)	3 hr/day (Mon-Fri) 6.5 hr/day (Sat)

Estimated Total Working Days: 90

Estimated Hours: 350 hours (This may change as per requirements).

## Essential Prerequisites

- I am able to run a single backend test target on my machine.

```
legion@pop-os: ~/Desktop/OpenSource/oppia
Tasks still running:
  core.controllers.editor_test (started 01:28:58)
-----
[datastore] Mar 14, 2022 1:29:01 AM io.gapi.emulators.grpc.GrpcServer$3 oper
ete
[datastore] INFO: Adding handler(s) to newly registered Channel.
[datastore] Mar 14, 2022 1:29:01 AM io.gapi.emulators.netty.HttpVersionRouti
channelRead
[datastore] INFO: Detected HTTP/2 connection.
20:00:03 FINISHED core.controllers.editor_test: 65.0 secs
Stopping Redis Server(name="sh", pid=382839)...
Stopping Cloud Datastore Emulator(name="sh", pid=382720)...

+-----+
| SUMMARY OF TESTS |
+-----+

SUCCESS   core.controllers.editor_test: 92 tests (61.6 secs)

Ran 92 tests in 1 test class.
All tests passed.

Done!
(P.3.7) legion@pop-os:~/Desktop/OpenSource/oppia$
```

- I am able to run all the frontend tests at once on my machine.

```
legion@pop-os: ~/Desktop/OpenSource/oppia
Headless 99.0.4844.51 (Linux x86_64): Executed 1125 of 7327 SUCCESS (0 secs / 12.811C
chrome Headless 99.0.4844.51 (Linux x86_64): Executed 1126 of 7327 SUCCESS (0 secs / 12
.824Chrome Headless 99.0.4844.51 (Linux x86_64): Chrome Headless 99.0.4844.51 (Linux x
86_64): Executed 1128 of 7327 SUCCESS (0 secs / 12.854 sChrChrome Headless 99.0.4844.51
(Linux x86_64): Executed 1130 of 7327 SUCCESS (0 secs / 12.88 secsChrome Headless 99
.0.4844.51 (Linux x86_64): Executed 1131 of 7327 SUCCESS (0 secs / 12.886 secChrome Hea
dless 99.0.4844.51 (LinuxChrome Headless 99.0.4844.51 (Linux x86_64): Executed 1135 o
f 7327 SUCCESS (0 secs / 12.934 secsChrome Headless 99.0.4844.51 (Linux x86_64ChromeCh
rome Headless 99.0.4844.51 (Linux x86_64): Executed 1151 of 7327 SUCCESS (0 secs Chrom
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 1155 of 7327 SUCCESS (0 secs / 1
3.059 secs) Chrome Headless 99.0.4844.51 (Linux xChrome Headless 99.0.4844.51 (Linux x
86_64): ExecuChrome HeadleChromChrome Headless 99.0.4844.51 (Linux x86_64): Executed 1
179 of 7327 SUCCESS (0 secs / 13.282 secs) Chrome Headless 99.0.4844.51 (Linux x86_64
): Executed 1182 of 7327 SUCCESS (0 secs / 13.338 secsChrome Headless 99.0.4844.51 (Li
nux x86_64): Executed 1184 of 7327 SUCCESS (0 secs / 13.35Chrome Headless 99.0.4844.51
(Linux x86_64): Executed 1190 of 7327 SUCCESS (0 secs /Chrome Headless 99.0.4844.51 (
Linux x86_64): Executed 1203 of 7327 SUCCESS (0 Chrome Headless 99.0.4844.51 (Linux x8
6_64): Executed 7327 of 7327 SUCCESS (2 mins 9.826 secs / 1 min 49.892 secs)
TOTAL: 7327 SUCCESS
TOTAL: 7327 SUCCESS
14 03 2022 00:38:36.738:WARN [launcher]: ChromeHeadless was not killed in 2000 ms, sen
ding SIGKILL.
Done!
(P.3.7) legion@pop-os:~/Desktop/OpenSource/oppia$
```

- I am able to run one suite of e2e tests on my machine.

```
legion@pop-os: ~/Desktop/OpenSource/oppia
[00:45:59] W/element - more than one element found for locator By(css selector, .toast
-success) - the first result will be used
. 🐞 should create, publish, check for thumbnail uploading error, unpublish and de
lete the blog post
[00:46:07] W/element - more than one element found for locator By(css selector, .toast
-success) - the first result will be used
[00:46:25] W/element - more than one element found for locator By(css selector, .toast
-success) - the first result will be used
. 🐞 should create multiple blog posts both published and drafts and check for nav
igation through list view

4 specs, 0 failures
Finished in 94.689 seconds

Executed 4 of 4 specs SUCCESS in 1 min 35 secs.
[00:46:32] I/launcher - 0 instance(s) of WebDriver still running
[00:46:32] I/launcher - chrome #01 passed
Stopping Protractor Server(pid=8755)...
Stopping Webdriver manager(name="sh", pid=8700)...
Stopping GAE Development Server(name="sh", pid=8436)...
[2022-03-14 00:46:32 +0530] [8952] [INFO] Handling signal: term
[2022-03-14 00:46:32 +0530] [8674] [INFO] Handling signal: term
```

## Other Summer Obligations

Currently I do not have any summer obligations.

## Communication Channels

I am active on Emails, Hangouts, and Discord. I can work with whatever platform that my mentor prefers. Meetings can be held every week to discuss progress in the project.

## Section 2: Proposal Details

### Problem Statement

<b>Link to PRD (or N/A if there isn't one)</b>	<b>N/A</b>
<b>Target Audience</b>	<ul style="list-style-type: none"><li>● Developers of the Oppia</li></ul>
<b>Core User Need</b>	<ul style="list-style-type: none"><li>● The project aims to migrate the exploration-editor page, State-editor, state-directives, question-directives, version-diff-visualization, contributor-dashboard-page, exploration-editor page, review test page, uiSortable, ngJoyride and practice session page from AngularJS to Angular for a better developer/user experience.</li></ul>
<b>What goals do we want the solution to achieve?</b>	<ul style="list-style-type: none"><li>● All of the files on the exploration-editor page, State-editor, state-directives, question-directives, version-diff-visualization, contributor-dashboard-page, exploration-editor page, review test page, uiSortable, ngJoyride, and practice session page must be migrated from angularJs to Angular.</li></ul>

## Section 2.1: WHAT

### Key User Stories and Tasks

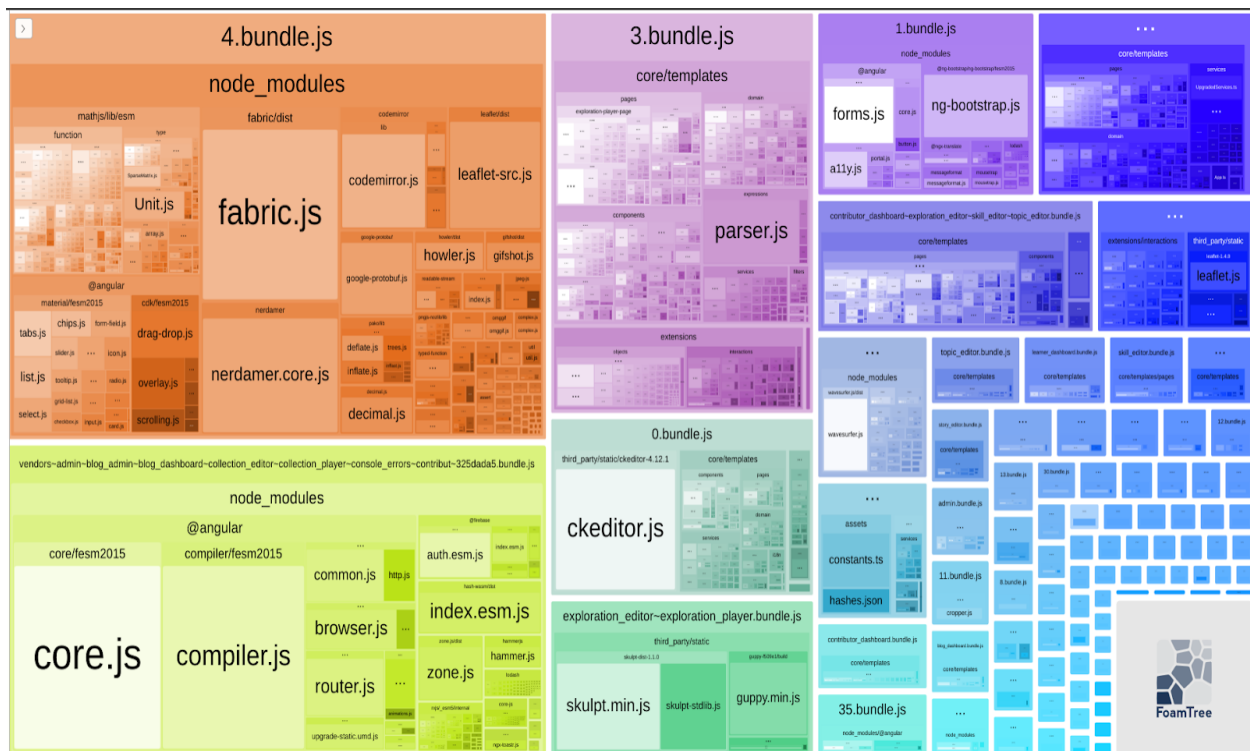
#	Title	User Story Description (role, goal, motivation) "As a ..., I need ..., so that ..."	Priority	List of tasks needed to achieve the goal (this is the "User Journey")	Links to mocks/prototypes, and/or PRD sections that spec out additional requirements.
1	Migration from angularJs to Angular11	As an Oppia Developer, I need to understand the Oppia codebase easily, so that I can start contributing.	HIGH	Read the wiki and only learn about Angular 8 (not about AngularJS).	N/A

## For User:

In the current system, we have already compiled the frontend code using webpack, and whenever users request any page the compiled code for that page is just served by the backend. these steps repeat whenever the user shifts to the new page. This process takes time to load the new page. When we completely migrate oppia from angularJs to angular. The user will be able to switch pages fast without any loading. After this project is completed, the end-user will be able to alter the application's view without having to reload. After migration from angularJS to angular, angularJs libraries are going to be removed which leads to a decrease in bundle size if the bundle size is going to decrease then the loading time, automatically going to decrease and we are also using lazy loading which makes the initial bundle size smaller. All these factors make the website load fast.

### Performance gain after migration:

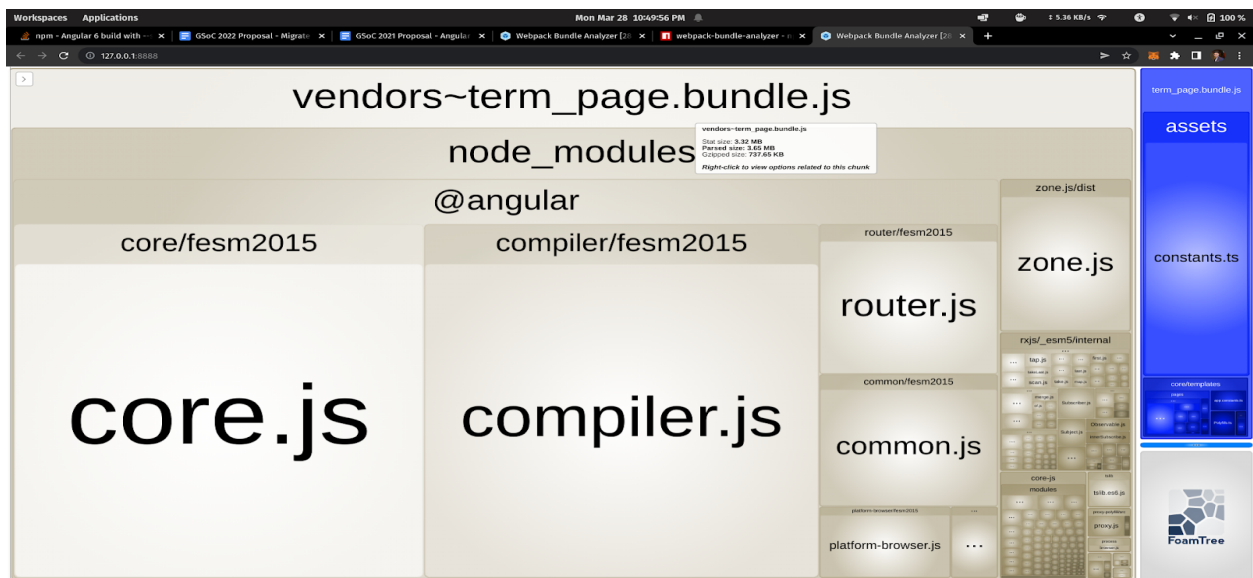
In the current build system every page like exploration-editor-page and contributor-dashboard-page is using separate node\_modules files, when we migrate from angularJs to angular all the pages routes are going to be unified i.e. there is common file duplication across every bundle is no longer remain. There is a reduction in common code that is directly proportional to a decrease in loading time.





To explain this I have taken an example of two pages before the unification of routes i.e. term\_page.module.ts and about\_page\_foundation.module.ts and created their bundle and analyzed them.

**NOTE:** I have used the [webpack-bundle-analyzer plugin](#) for this information.

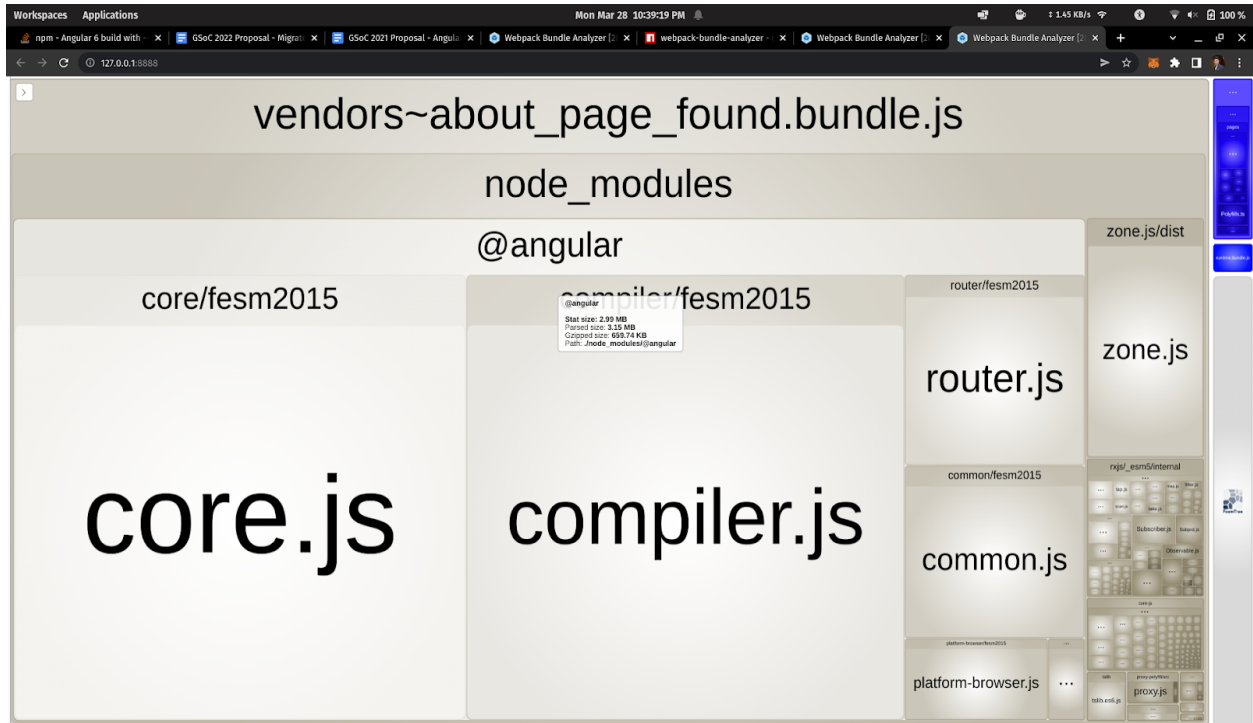


### Gzipped sizes

Total size of TermPage bundle: 737.65kb

Size of actual code for term page: 12.54kb

Percentage of common code/ node libraries in this module is: 98.3%



### Gzipped sizes

Total size of TermPage bundle: 737.65kb

Size of actual code for term page: 7.07kb

Percentage of common code/ node libraries in this module is: 99%

When Both Pages are unified through routing, common libraries are only imported once, and duplication of the file is no longer occur.



Total size gzipped of both pages: 712.29kb  
 Size of actual code (term + about page is): 35.8kb

When both bundles of pages are unified, **we find a 52 percent reduction in code size**. In these examples, we have taken small pages into the account. There are more pages in the oppia project where node\_modules alone gzipped size is more than 2-3 MB. after the unification of all the pages into one there is a huge reduction of common code. In the above case, the 52 percent of code is reduced which means the user has to fetch 52 percent less code which automatically is going to reduces the loading time by 52 percent. By lazy loading we can reduce the initial bundle size and loading time more.

## Section 2.2: HOW

I have divided this project into two parts.

1. Migration of State-editor, state-directives, question-directives, version-diff-visualization, contributor-dashboard-page.
2. Migration of exploration-editor page, the review test page, the practice session page, from AngularJS to Angular 11. Migration of the relevant files in the codebase to use AngularCDK DragAndDrop instead of uiSortable and ngxJoyride instead of ngJoyride. Remove uiSortable and ngJoyride completely from the codebase.

### Migration from angularJs to angular.

Migration of Directives/Controllers/Services can be done by following this approach.

- First, we have to check whether this component/service is depending upon some other. If so, then we have to migrate that dependent component/service first. If not then we are good with the migration of that component/service.
- Changing file name from .directive / .controller to .component.
- Changing ES5 imports to ES6 imports. i.e. from *require('service/xyz');* to *import{ Migration of the relevant files in the codebase to use AngularCDK DragAndDrop instead of uiSortable and ngxJoyride instead of ngJoyride. Remove uiSortable and ngJoyride completely from the codebase.xyzComponent} from 'service/xyz';*
- All the services required by the component should be injected in the component/service i.e. known as [dependency injection](#).

Example: constructor(  
private logger: Logger  
) { }

- Declaration of all variables with @Input() or @Output() prefix present inside bindToControllers.

Example:

```
bindToController: {  
  value: '='  
}
```

```
@Input() value: string;
```

- Then migrate the component using the [angular.io](#)
- In angularJs file update data binding at calling part of that migrated component i.e.

local-value=someValue to `[local-value]=somevalue` (Kebab Case).

And in the angular file change local-value=somevalue to [localValue]=somevalue (Camel Case).

- Then finally I will downgrade the component by using [downgradeComponent](#). example:

```
angular.module('oppia').directive( 'xyz', downgradeComponent( {component: xyzComponent}));
```

### Example of Migration of third-party library:

Migration of ngJoyride to [@ngx-joyride](#) in angular

For migrating this i am going to follow this steps:

1. Then in exploration-editor-page.module.ts, I am going to import the JoyrideModule.

```
import { JoyrideModule } from 'ngx-joyride';
```

```
@NgModule({
  imports: [
    JoyrideModule.forRoot(),
    BrowserModule,
  ],
})
```

2. Then I will add the following code in the desired component .ts file.

```
import { JoyrideService } from 'ngx-joyride';
```

```
constructor(
  private joyride: JoyrideService
){}

start(): void {
  this.joyride.startTour(
    { steps: ['animation1', 'animation2', 'animation3', 'animation0'] }
  );
}
```

3. Then, i will add the following code in the desired component .html file.

```
<h1 joyrideStep="animation0" [stepContent]="customContent0">Animation page step 0</h1>
```

```
<h1 joyrideStep="animation1" [stepContent]="customContent
```

```
1">Animation page step 1</h1>
```

```

<h1 joyrideStep="animation2" [stepContent]="customContent2">Animation page step 2</h1>

<h1 joyrideStep="animation3" [stepContent]="customContent3">Animation page step 3</h1>

<ng-template #customContent1>
  #customContent1
</ng-template>
<ng-template #customContent2>
  <h1>#customContent2</h1>
</ng-template>
<ng-template #customContent3>
  #customContent3
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Cumque et quas quam! Omnis error ut
  sequi iusto voluptates, facere quaerat.</p>
</ng-template>
<ng-template #customContent0>
  #customContent0
</ng-template>

<button class="btn btn-primary shivamBtn" (click)="start()">start animation again</button>

```

4. You can find [Demo Video](#) here.

I am planning to migrate in three steps to reduce the complexity while migrating.

1. ion
2. Component/Directive Services migratmigration
3. Routing Implementation & Removal of angularJs code/Code cleaning.

( [existing AngularJs to Angular migration PRs](#) )

## Files Under Milestone 1:

Milestone 1 Files	
File name	Total no. of lines
parameterize-rule-description.filter.ts	369
add-answer-group-modal.controller.ts	284
rule-type-selector.directive.ts (select2.import.ts)	88
html-select.component.ts	62
rule-editor.component.ts	706
outcome-editor.component.ts	619
training-modal.controller	635
parameter-metadata.service.ts	484
exploration-warnings.service.ts	1917
training-panel.component.ts	309
training-data-editor-panel-modal.controller.ts	678
training-modal.service.ts	119
training-data.service.ts	511
test-interaction-panel.component.ts	137
truncate-input-based-on-interaction-answer-type.filter.ts	25
training-data-editor-panel.service.ts	100
answer-group-editor.component.ts	829
state-responses.component.ts	1810
state-editor.component.ts	351
question-undo-redo.service.ts	9
question-editor-modal.controller.ts	675
question-update.service.ts	721
question-editor.component.ts	634
question-player-concept-card-modal.controller.ts	169
skill-mastery-modal.controller.ts	73
convert-html-to-unicode (remove angular.element)	46
convert-unicode-to-html.filter.ts	62
convert-unicode-with-params-to-html	278
question-player.component.ts	1151

skill-editor-routing.service.ts	120
question-creation.service.ts	857
translation-status.service.ts	986
questions-list-select-skill-modal.controller.ts	73
questions-list.component.ts	1755
router.service.ts	745
state-graph-visualization.directive.ts	914
version-diff-visualization.component.ts	487
question-suggestion-review-modal.controller.ts	689
contributions-and-review.component.ts	1710
question-suggestion-editor-modal.controller.ts	579
question-opportunities.component.ts	496
contributor-dashboard-page.component.ts	447
<b>Total line to migrate in milestone 1:</b>	<b>23709</b>

### Files Under Milestone 2:

<b>MileStone 2 Files</b>	
<b>File name</b>	<b>Total no. of lines</b>
editor-navbar-breadcrumb.component.ts	154
editor-navigation.component.ts	684
exploration-graph.component.ts	253
training-modal.controller.ts	635
state-name-editor.component.ts	428
exploration-graph-modal.controller.ts	128
teach-oppia-modal.controller.ts	514
training-unresolved-answer-modal.template.html	20
training-data-editor.template.html	100
unresolved-answers-overview.component.ts	312
exploration-editor-tab.component.ts	1501
exploration-objective-editor.component.ts	48
exploration-save-and-publish-buttons.component.ts	600
exploration-title-editor.component.ts	132
feedback-tab.component.ts	750
compare-versions.service.ts	1498



history-tab.component.ts	755
improvement-modal.service.ts	126
improvements-tab.component.ts	439
needs-guiding-responses-task.component.ts	74
exploration-metadata-modal.controller.ts	349
exploration-save-modal.controller.ts	73
param-changes-editor.component.ts	690
value-generator-editor.directive.ts	86
preview-tab.component.ts	600
exploration-automatic-text-to-speech.service.ts	26
exploration-save.service.ts	1314
settings-tab.component.ts	1787
pie-chart.component.ts	257
statistics-tab.component.ts	393
exploration-editor-suggestion-modal.controller.ts	329
suggestion-modal-for-exploration-editor.service.ts	339
audio-translation-bar.directive.ts	1319
Add-audio-translation-modal.controller.ts	186
voiceover-recording.service.ts	152
state-translation.component.ts	1912
state-translation-editor.component.ts	685
state-translation-status-graph.component.ts	264
translator-overview.component.ts	358
translation-tab.component.ts	957
exploration-editor-page.component.ts	1414
review-test-page.component.ts	279
practice-session-page.component.ts	270
<b>Total line to migrate in milestone 1:</b>	<b>23190</b>

## Process for verifying that the migration causes no breakages

- Firstly, I will migrate the stuff to the best of my knowledge
- Then I will test it too. (for example, if I am migrating the setting tab in the exploration editor page then I will test that all the settings are working properly as in oppia.org)
- If there was a major change coming in my PR then I will inform the Q/A team beforehand to take their help so that we can test this PR earlier before merging it to the develop branch
- Proper reviews.
- The end-to-end and lighthouse tests will ensure that the main functionality of the component is always going to work.  
(This is the standard procedure to avoid migration breakages).

## Any potential blockers:

- No

**Note:** While making the proposal, I have taken the reference from [Angular Doc](#) and [2021 GSoC Proposal](#) made by [Ashutosh Chauhan](#) .

## Testing Plan

This project requires some test runs on the prod server to make sure the new build system works well.

### E2e testing plan

There is no need to add any e2e tests.

### Feature testing

Does this feature include non-trivial user-facing changes?

**NO**

# Implementation Plan

## Milestone 1 Table:

Fully migrate the following directives/pages from AngularJS to Angular 11 in milestone1:

1. State-editor
2. state-directives
3. question-directives
4. version-diff-visualization
5. contributor-dashboard-page

No.	Description of PR / action	Prereq PR numbers	Target date for PR creation	Target date for PR to be merged
1	parameter-metadata.service.ts exploration-warnings.service.ts training-modal.service.ts router.service.ts training-data.service.ts training-data-editor-panel.service.ts question-undo-redo.service.ts question-update.service.ts	N/A	20 June	25 June
2	skill-editor-routing.service.ts question-creation.service.ts translation-status.service.ts parameterize-rule-description.filter.ts add-answer-group-modal.controller.ts rule-type-selector.directive.ts (select2.import.ts) html-select.component.ts rule-editor.component.ts	N/A	27 June	2 July
3	outcome-editor.component.ts training-modal.controller training-panel.component.ts training-data-editor-panel-modal.controller.ts test-interaction-panel.component.ts truncate-input-based-on-interaction-answer-type.filter.ts answer-group-editor.component.ts	N/A	4 July	9 July
4	state-responses.component.ts state-editor.component.ts <b>(State-editor / State-Directive migration)</b> question-editor-modal.controller.ts question-editor.component.ts question-player-concept-card-modal.contr	N/A	11 July	16 July

	oller.ts skill-mastery-modal.controller.ts			
<b>5</b>	convert-html-to-unicode (remove angular.element) convert-unicode-to-html.filter.ts convert-unicode-with-params-to-html question-player.component.ts questions-list-select-skill-modal.controller.ts questions-list.component.ts <b>(question-directive Migration)</b>	N/A	18 July	23 July
<b>6</b>	state-graph-visualization.directive.ts version-diff-visualization.component.ts <b>(version-diff-visualization Migration)</b> question-suggestion-review-modal.controller.ts contributions-and-review.component.ts	N/A	25 July	30 July
<b>7</b>	<b>question-suggestion-editor-modal.controller.ts</b> <b>question-opportunities.component.ts</b> <b>contributor-dashboard-page.component.ts</b> <b>(contributor-dashboard-page Migration)</b>	N/A	1 August	5 Aug
<b>8</b>	<b>Milestone 1: Buffer time for Completion</b>	N/A	1Aug - 8Aug	Nil

## Milestone 2 Table:

Fully migrate the exploration-editor page, the review test page, the practice session page, from AngularJS to Angular 11. Migrate the relevant files in the codebase to use AngularCDK DragAndDrop instead of uiSortable and ngxJoyride instead of ngJoyride. Remove uiSortable and ngJoyride completely from the codebase.

<b>No.</b>	<b>Description of PR / action</b>	<b>Prereq PR numbers</b>	<b>Target date for PR creation</b>	<b>Target date for PR to be merged</b>
<b>1</b>	editor-navbar-breadcrumb.component.ts editor-navigation.component.ts exploration-graph.component.ts state-name-editor.component.ts exploration-graph-modal.controller.ts teach-oppia-modal.controller.ts training-unresolved-answer-modal.template.html training-data-editor.template.html	Milestone 1 needs to be merged	17 August	22 Aug

2	training-modal.controller.ts unresolved-answers-overview.component.ts exploration-editor-tab.component.ts <b>(ngJoyride / uiSortable)</b> exploration-objective-editor.component.ts exploration-save-and-publish-buttons.component.ts exploration-title-editor.component.ts feedback-tab.component.ts	N/A	26 August	1 sep
3	compare-versions.service.ts history-tab.component.ts improvement-modal.service.ts improvements-tab.component.ts needs-guiding-responses-task.component.ts exploration-metadata-modal.controller.ts exploration-save-modal.controller.ts param-changes-editor.component.ts value-generator-editor.directive.ts preview-tab.component.ts exploration-automatic-text-to-speech.service.ts exploration-save.service.ts	N/A	3 Sep	9 Sep
4	settings-tab.component.ts pie-chart.component.ts statistics-tab.component.ts exploration-editor-suggestion-modal.controller.ts suggestion-modal-for-exploration-editor.service.ts audio-translation-bar.directive.ts Add-audio-translation-modal.controller.ts voiceover-recording.service.ts state-translation.component.ts	N/A	13 Sep	20 Sep
5	state-translation-editor.component.ts state-translation-status-graph.component.ts translator-overview.component.ts translation-tab.component.ts <b>exploration-editor-page.component.ts</b> <b>review-test-page.component.ts</b> <b>practice-session-page.component.ts</b>	N/A	23 Sep	30 Sep
6	<b>Milestone 2: Buffer time for Completion or e2e resolve</b>	N/A	October 3	N/A

## Future Work

**Note:** This section is mainly for reference (since it is understood that items in this section will not be part of the GSoC project). Proposals will primarily be evaluated based on the implementation plan above.

### **Angular CLI Addition:**

(Note: Angular CLI addition is not in the scope of this GSoC project).

In the future, I am going to add angular CLI with ahead-of-time (AOT) compilation and removal of webpack from the project.

At present webpack builds applications for each page separately. After migration of the angularJs to angular all the entry points will be unified. Then I will add angular CLI to the build system, when it is fully integrated with the build system, I will start removing the webpack from the project.

### **Advantages of using Webpack:**

It allows you greater control over the development process, and webpack is entirely customizable.

### **Cons of using Webpack:**

It takes a lot of knowledge to configure it right, and Angular CLI is easier to implement than webpack.

### **Pros of using Angular CLI:**

It is easier to understand than webpack, and it supports server-side rendering as well. It does not require much configuration or boilerplate code, it also optimizes the build files, and it also supports AOT Compilation.

### **Cons of using Angular CLI:**

- Less customizable than webpack. It can be overcome by using the [@angular-builders/custom-webpack](https://github.com/angular-builders/custom-webpack) plugin to achieve customizability over the build process.

[\(Why AngularCLI can't be implemented before migration to Angular?\)](#)