

Section 2: About Your Project

Project Details

Project title	Contributor Dashboard Admin Stats Table
Project size	Medium (~175 hours)
Why did you choose this project?	<p>I have been actively collaborating with the LaCE team for a considerable period, which has made me familiar with the codebase. However, my interest in Oppia's code extends beyond the frontend, and I have a keen desire to expand my knowledge of the backend as well.</p> <p>This year, I chose to focus my GSoC project on the contributor dashboard, as I am fascinated by the entire lifecycle of submitting a question or translation. I am eager to delve deeper into this aspect of the codebase and gain more comprehensive knowledge of the software as a whole.</p> <p>One of my primary objectives for this project is to add a new endpoint to the contributor dashboard admin controller. I believe that this experience will give me valuable exposure to the backend code and enable me to learn more about how the application functions behind the scenes. Additionally, this will be a significant challenge for me, and I am excited to develop my problem-solving and coding skills as I work to implement this new feature.</p> <p>Overall, I am thrilled to continue my journey with Oppia and to explore more of the codebase through this year's GSoC project. I look forward to the opportunities and challenges that lie ahead and am eager to learn as much as I can from this experience.</p>

Project Timeframe

Note: Oppia will only be offering a single GSoC coding period timeframe this year, starting on **May 29**. All work for Milestone 1 must be completed and submitted by **July 14**, and all work for Milestone 2 must be completed and submitted by **Sept 15**. We will not be able to extend these deadlines.

Coding period	<ul style="list-style-type: none">I will adhere to the above deadlines.
Planned time commitment	May 29 to July 15 :- 6-8 hours/day, 1 holiday/week. July 15 to Sept 15 :- 3-4 hours/day, 1 holiday/week. Approx 4 miscellaneous holidays/month.
What other obligations might you need to work around during the summer?	I'll be having my summer vacation from May 29 to July 15 where I can dedicate my maximum time to the project. From July 15 onwards with my classes going on I can work approx 15-20 hrs/week.

Communication Channels

Note: The Oppia team places a high emphasis on communication, and we have found that daily contact between contributors and mentors is important for helping keep projects on track. This is why we ask that contributors send short daily updates to their mentors explaining what they have done, where they are stuck, and what they plan to do next.

I can commit to sending daily updates to my mentor by email, each day I work during the GSoC period.	<ul style="list-style-type: none">• Yes
In addition to the above: how often, and through which channel(s), do you plan on communicating with your mentor?	<ul style="list-style-type: none">• Weekly sync meetings to give updates or discuss target for the upcoming weeks as well as discuss the problems faced in previous weeks work• A google doc to track the progress of the project

Section 3: Proposal Details

Problem Statement

Link to PRD (or N/A if there isn't one)	Admin Dashboard PRD
Target Audience	<p>Web Super Admin: The one who is in charge of Oppia's translation team. They assign language coordinators. They also request the tech team to add new languages to the contributor dashboard that are not already present.</p> <p>CD language coordinator: they're dedicated volunteers from around the world selected to coordinate translations and are responsible for controlling the review access for translators for the language assigned to them.</p> <p>CD practice question Coordinator: they are appointed by the web super admin, and are responsible for appointing practice questions submitters and reviewers.</p>
Core User Need	<p>Web Super Admin</p> <p>As a Web Super admin I need to assign language coordinators and give them access to CD for the assigned language so that they could then add or remove reviewers.</p>

	<p>CD language coordinator As a CD language coordinator, I need to easily track and compare the progress of volunteers, and see their accuracy to get candidates for reviewer roles.</p> <p>CD practice questions coordinator As a CD practice questions admin, I need to easily track and compare the progress of volunteers, and see their accuracy to get candidates for reviewer roles.</p>
What goals do we want the solution to achieve?	<p>Improve discoverability of candidates for the “reviewer” role Improve efficiency in accessing volunteer data and managing volunteer contribution rights</p> <p>Prevent inadvertent changes to volunteer settings by restricting access based on user role</p>

Section 3.1: WHAT

This section enumerates the requirements that the technical solution outlined in “Section 2: HOW” must satisfy.

Key User Stories and Tasks

#	Title	User Story Description (role, goal, motivation) “As a ..., I need ..., so that ...”	Priority ¹	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	Add language coordinator	As a Web Super Admin, I need to assign language coordinators and give them access to CD for the assigned language so that they could then add or remove reviewers.	Must have	<ul style="list-style-type: none"> • Login to the /contributor-dashb oard-admin. • Go to the language coordinator tab. • Click on “add volunteer” button • Enter oppia username of the volunteer • Select languages 	Add Language Coordinator Modal

¹ Use the **MoSCow** system (“Must have”, “Should have”, “Could have”). You can read more [here](#).

				to be assigned <ul style="list-style-type: none"> Click invite 	
2	Promote a text translator to reviewer	.As a language coordinator, I need to track the progress of translation volunteers so that I can get candidates for reviewers and promote good translators to reviewers.	Must have	<ul style="list-style-type: none"> Login to the /contributor-dashb oard-admin Go to the language translators tab. Click on edit option Check the "translation reviewer" role Click save and confirm. 	Edit Role - Translation reviewer
				Similar approach to demote: <ul style="list-style-type: none"> In the edit modal uncheck the role to demote. 	
3	Promote translator/ reviewer to be language coordinator	As a Web Super admin, I need to promote text translators/review ers to be language coordinators so that they could then manage the assigned language.	Must have	<ul style="list-style-type: none"> Login to the /contributor-dashb oard-admin. Click on edit option of the user Check the "language coordinator" role Click save and confirm. 	Edit Role - Add coordinator Confirmation Modal
				Similar approach to demote: <ul style="list-style-type: none"> In the edit modal uncheck the role to demote. 	
4	Add practice questions submitter	As a practice questions admin, I need to add question submitters and give them access to CD so that they could then start submitting questions.	Must have	<ul style="list-style-type: none"> Login to the /contributor-dashb oard-admin. Go to the question submitter tab. Click on "add volunteer" button Enter oppia username of the volunteer Click add 	Add Question Submitter
5	Promote practice questions submitter	.As a practice questions admin, I need to track the progress of	Must have	<ul style="list-style-type: none"> Login to the /contributor-dashb oard-admin. Go to the language 	Promote Question submitter to question reviewer

	to become practice questions reviewer	question submitters so that I can promote good question submitters to reviewers.		<p>question submitter tab.</p> <ul style="list-style-type: none"> • Click on edit option • Check the "question reviewer" role • Click save and confirm. <p>Similar approach to demote:</p> <ul style="list-style-type: none"> • In the edit modal uncheck the role to demote. 	
--	---------------------------------------	---	--	---	--

Technical Requirements

Additions/Changes to Web Server Endpoint Contracts

- I have added all the handlers that will be used rather than adding only the handlers that will be modified. Some of the handlers will be used as it is and will not be modified/changed.

#	Endpoint URL	Request type (GET, POST, etc.)	New / Existing	Description of the request/response contract (and, if applicable, how it's different from the previous one)
1.	/<contribution_type>/<contribution_subtype>/<username>/<contributorstatssummaries Link to Existing code	GET	Existing	The existing handler checks for only if the user can access the contributor dashboard by @acl_decorators.can_fetch_contributor_dashboard_stats. One more filter will be added where it returns the contribution stats of only that language for which coordinator has permission.
2.	/getcontributorusershandler/<category> Link to Existing	GET	Existing	This handler will be used as it is without any changes.

	code			
3	/contributor/rightshandler/<category> > Link to Existing Code	POST	Existing	Will be used as it is to update the rights of a contributor.
4	/languagecoordinator/rightshandler/<category>	POST	New	<p>This handler will be used to give a user, or withdraw from a user, the language coordinator role for a particular language. And param to be passed is</p> <pre> username: { 'schema': { 'type': 'basestring' } }, 'action': { 'schema': { 'type': 'basestring', 'choices': ['add', 'remove'] } }, 'language_code': { 'schema': { 'type': 'basestring' } } </pre>
5	/contributor/rights/data/handler Link to Existing Code	GET	Existing	<p>Calling this endpoint will show the contribution rights of the user.</p> <pre> params: { username: string } </pre>
6	/allcontributorstats/summaries/<category>/<sub_category>	GET	New	<p>The response will be sequence of the stats model for eg. if it's for Category :- translation Sub-category :- contribution, then response will be Sequence[TranslationContributionStatsModel] You can see the docstring here.</p> <p>Possible options:- Category:- translation/question Sub-category:- contribution/review</p>

Calls to Web Server Endpoints

#	Endpoint URL	Request type (GET, POST, etc.)	Description of why the new call is needed, or why the changes to an existing call is needed
1.	<contribution_type>/<contribution_subtype>/<username>/contributionstatssummaries Link to Existing code	GET	The existing handler checks for only if the user can access the contributor dashboard by @acl_decorators.can_fetch_contributor_dashboard_stats. One more filter will be added where it returns the contribution stats of only that language for which coordinator has permission.
2.	/getcontributoruserhandler/<category> Link to Existing code	GET	Calling this endpoint will provide a list of all the contributor users in list format. The request will contain the category of the contribution. params = { language_code: languageCode }
3	/contributionrightshandler/<category> Link to Existing code Link to Existing Code	POST	Calling this endpoint will help in adding or removing contribution rights of a particular user. The request will contain the category of the contribution params: { username: string; language_code: string; }
4	/adminrolehandler Link to Existing Code	POST	Update the role of Question Coordinator.
5	/languagecoordinatorrights/	POST	We can use the existing handler (AdminRoleHandler) to update the role of question coordinator but for language coordinator we need to assign a particular language(analogous to how we do for topic manager). Currently we have different handler (TopicManagerRoleHandler) for managing topic related rights.
6	/contributionrightsdatahandler Link to Existing Code	GET	It will be used to get the contribution rights of the user.
7	/allcontributorstatssummaries/<category>/<sub_category>	GET	Calling this endpoint will give the stats of contribution of a particular category and sub_category.

UI Screens/Components

#	ID	Description of new UI component	i18n required?	Mock/spec links	A11y requirements
1.	Contributor Dashboard Admin Page Component	<p>Contributor Dashboard admin page should now have a tabular view to view and manage all the contributors. It'll have following 5 tabs :-</p> <ul style="list-style-type: none"> Text Translators <ul style="list-style-type: none"> User Performance Accuracy Translated Cards Last Translated Role(EDIT option) Translation Reviewers <ul style="list-style-type: none"> User Reviewed Cards Last Reviewed Role(EDIT option) Language Coordinators <ul style="list-style-type: none"> Language Coordinators Reviewers Translators Question Submitters <ul style="list-style-type: none"> User Performance Acceptance Questions Submitted Last Submitted Role(EDIT option) Question Reviewers <ul style="list-style-type: none"> User Questions Reviewed Last Reviewed Role(EDIT option) <p>*Sub-bullets means contents of that tab in the table</p>	No	<p>Text Translators tab Mock</p> <p>Transaltion Reviewers tab Mock</p> <p>Language Coordinator tab Mock</p> <p>Question Submitters tab Mock</p> <p>Question Reviewer tab Mock</p>	Elements will have focus management with tab and ARIA attributes for easy keyboard interaction.
2.	Filter Component	<p>Filter for contributors stat. Two filters will be provided they are as follows:-</p> <ul style="list-style-type: none"> Topic <ul style="list-style-type: none"> Multi-select Language <ul style="list-style-type: none"> Single-select 	No	Mock	Dropdown menu will have proper tab attributes for keyboard interaction and focus management.
3	Table Component	<p>Use to display all the contributors in table format with following columns</p> <ul style="list-style-type: none"> Content will vary depending upon category of the stats 	No	Mock	Each entry in a row will be marked with

					appropriate ARIA attributes so that it can be perceived by screen readers easily.
4	Edit Role Modal	Modal will be overlaid on top of the page and can be dismissed by the cancel button. Contents of the edit role will differ in different situations.	No	Mock	Each button and input field will have ARIA labels.

Data Handling and Privacy

#	Type of data	Description	Why do we need to store this data?	Anonymized?	Can the user opt out?	Wipeout policy	Takeout policy
1.	Performance (Translation Contribution StatsModel, QuestionContributionStatsModel)	Number of accepted/rejected contributions in latest 100 contributions	We need this data to measure the performance of the user. This data can be used by admins to decide whether to promote contributors to reviewers or not.	No	No	If a user deletes the account, this data will be deleted	User can export this data while exporting the Contribution Stats Model
2	User Contribution Rights (UserContributionRightsModel)	Keeps track of if the user is a coordinator or not.	We need this so we can promote a contributor to coordinator and to allow right access to right user.	No	No	If a user deletes the account, this data will be deleted	User can export this data while exporting the Contribution Stats Model

Other Requirements

N/A

Section 3.2: HOW

Existing Status Quo

- The current state of the Contributor admin dashboard design is one that has not been updated since its creation. As it stands, the admin is burdened with the tedious task of entering the username of the contributor every time they want to perform basic tasks. The admin is solely reliant on the reviewers to provide recommendations for potential translators who could become reviewers. Furthermore, the admin must then cross-check the accuracy of these translators by delving into their stats.
- It's a rare occasion that the admin revokes the rights of a reviewer, but it does happen, mostly when the reviewer fails to respond to follow-up emails. At present, the admin is forced to request progress updates from the coordinators of different languages, with no visual representation of the progress of each language. Additionally, anyone with access to the admin dashboard can manage contribution rights, leading to potential security issues. Even more complex, to add a new language to the language drop-down, the admin must send an email request to the Contributor experience development team, which is time-consuming and inefficient.
- In short, the current state of the admin dashboard design is far from optimal, and it is essential that necessary improvements are made to ensure the smooth running of administrative tasks.

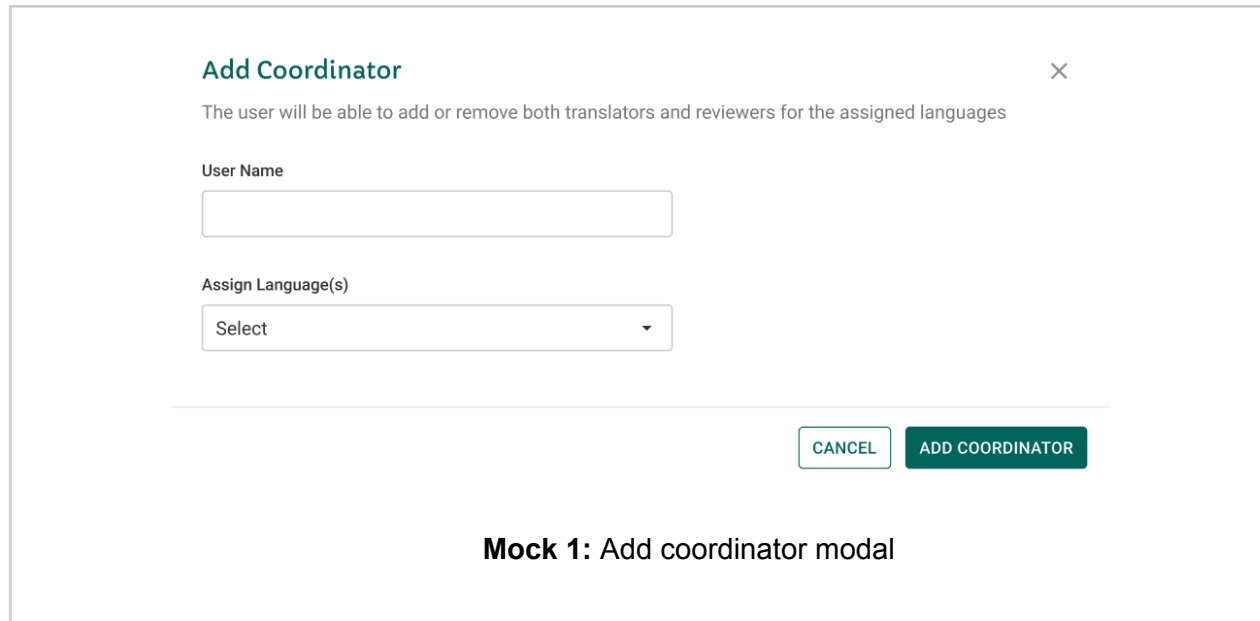
Solution Overview

- **Limited Access to views**

Views/ Roles	Text Translators	Translation Reviewers	Language Coordinators	Question Submitters	Question Reviewers	Question Coordinators
Web Super Admin	✓	✓	✓	✓	✓	✓
Language Coordinator (only for assigned languages)	✓	✓				
CD Practice Questions Coordinator				✓	✓	

1. Add language coordinator

The CD language admin can add users as language coordinators for specific languages using their oppia username.



The image shows a modal window titled "Add Coordinator" with a close button (X) in the top right corner. Below the title is a subtitle: "The user will be able to add or remove both translators and reviewers for the assigned languages". There are two input fields: "User Name" with a text input box, and "Assign Language(s)" with a dropdown menu showing "Select". At the bottom right, there are two buttons: "CANCEL" and "ADD COORDINATOR".

Mock 1: Add coordinator modal

The coordinator will be able to do the following for the assigned languages:

- view volunteer progress
- add/remove contribution rights volunteers
- Mark volunteers as inactive

Coordinators can be assigned to languages enabled for the Contributor Dashboard (as determined by Contributor Dashboard Admins).

2. Promote a Text Translator to Translation Reviewer

The Web Super admin and language coordinator can promote a text translator to Translation reviewer by comparing their stats in the tabular view. The translators are sorted based on their performance score.

Table Attributes

Performance score: Performance score for text translators will be based only on the last 100 cards submitted by the translator. Each accepted card will give one point increase in the performance while each rejected card will decrease performance by two points. It will range from -200 to 100.

performance score = accepted cards - 2 (rejected cards)

note: only last 100 cards will be considered for this calculation

Accuracy: accepted cards/ submitted cards

Translated cards: submitted cards

Last translated: submission time of last submitted card

Filters

Language: Only one of the assigned languages could be selected. By default the languages assigned to the coordinator will be selected. On the first visit, the language which comes alphabetically first will be selected.

We can use the localStorage feature to store a key value pair just like we do to save the user's preferred language.

```
this.localStorage.setItem(contributorAdminLastSelectedLanguage, language_code);
```

This key-value pair will be retrieved when the user revisits the contributor-dashboard-admin.

Topic: one or more topics could be selected. By default none are selected. When none is selected all the contributors of a particular language will be displayed. When we apply the topic filter, the contributors who have made at least one contribution to that topic will be displayed.

Last translated: all time, within 7 days, within 30 days and within 90 days would be options. By default, "all time" would be selected.

Admin Dashboard

Translators

24

Reviewers

8

Coordinators

6

TEXT TRANSLATORS

TRANSLATION REVIEWERS

LANGUAGE COORDINATORS

Filter

Language Hindi ▾

Topics Select ▾

Last Translated Within 90 days ▾

+ ADD TRANSLATOR

User	Rank	Performance	Accuracy	Translated Cards	Last Translated	Role
> crestedoutsiders	1	97	98.5%	208	1 day ago	EDIT
> giglio	2	96	99%	198	4 days ago	EDIT
> jgoobb	3	90	100%	190	5 days ago	EDIT
> sprouts	4	88	100%	204	2 days ago	EDIT
> zhi_117	5	87	92%	187	6 days ago	EDIT
> 247_coder	6	86	88%	177	19 days ago	EDIT
> alvariz_	7	80	100%	123	20 days ago	EDIT
> mdogtto	8	77	50%	228	45 days ago	EDIT
Showing 1- 20 of 24 Results			< 1 2 >		Results per page 20 ▾	

Mock 2.1: Tabular view

Admin Dashboard

Translators
24

Reviewers
8

Coordinators
6

TEXT TRANSLATORS

TRANSLATION REVIEWERS

LANGUAGE COORDINATORS

Filter

Language Hindi

Topics 2 Selected

Last Translated Within 90 days

+ ADD TRANSLATOR

User	Rank	Performance	Accuracy	Translated Cards	Last Translated	Role
▼ crestedoutsiders	1	97	100%	204	1 day ago	EDIT
<div>Date JoinedMar 10, 2021</div> <div>LanguagesHindi, Dari</div> <div>Accepted Translations22 Cards, 240 Words (20 Cards without Edits)</div> <div>Submitted Translations24 Cards, 300 Words</div> <div>Rejected Translations2 Cards, 60 Words</div> <div>Active TopicsPlace Values, Addition & Subtraction, Multiplication</div>						
> rengut	2	96	98.5%	208	4 days ago	EDIT
> giglio	3	90	99%	198	5 days ago	EDIT
> jgoobb	4	88	100%	190	2 days ago	EDIT
> sprouts	5	87	92%	187	6 days ago	EDIT
Showing 1-20 of 24 Results						
< 1 2 >						
Results per page 20						

Mock 2.2: Expanded row

Edit Role for <username>

×

User Role

☐ Translation Reviewer

Reviews content translated by volunteers

☒ Language Coordinator

Helps manage and coordinate volunteers

Assigned Languages

Hinglish, Hindi

CANCEL

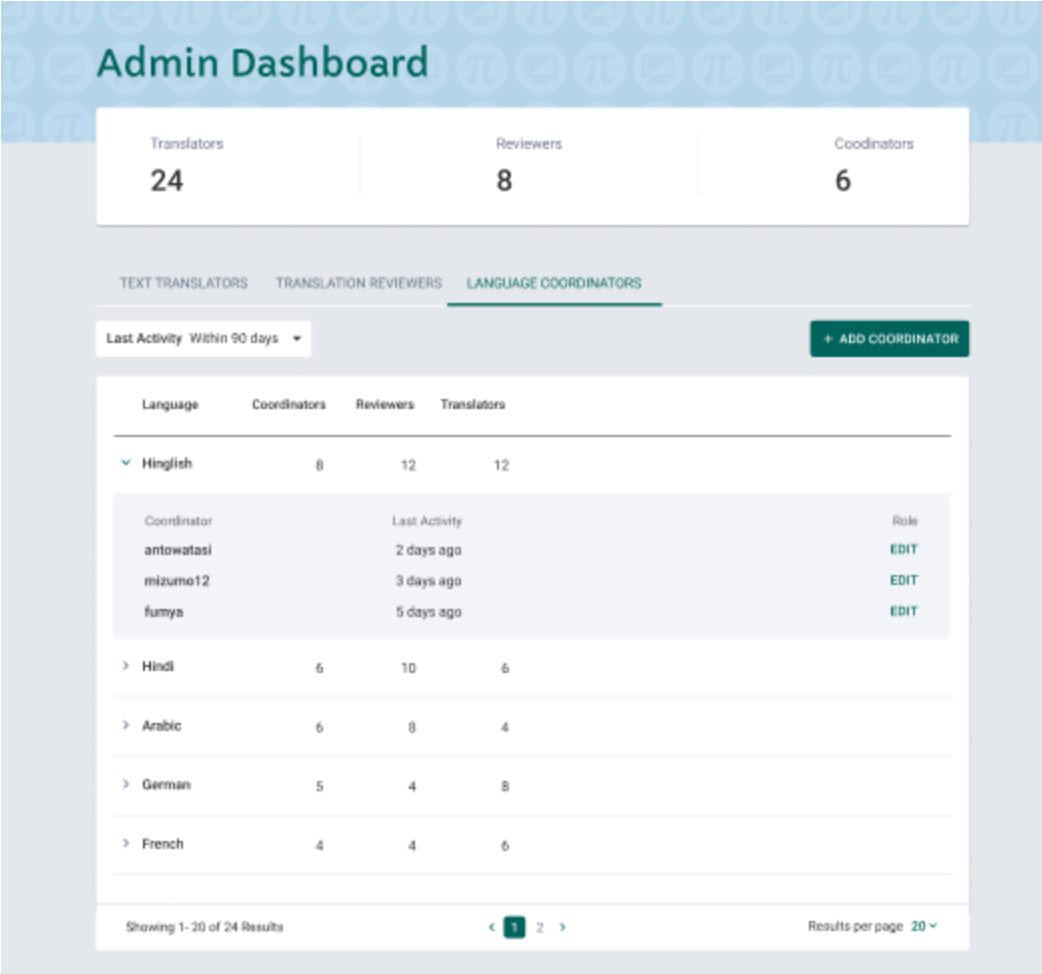
SAVE CHANGES

3. Promote Translation Translator/Reviewer to Language Coordinator

This action can only be performed by the web super admin.

Last activity: time stamp for action to add/remove reviewers

All languages with at least one volunteer will be listed here.



Mock 2.1: Tabular view

Edit Role for <username>

User Role

☐ Translation Reviewer

Reviews content translated by volunteers

☒ Language Coordinator

Helps manage and coordinate volunteers

Assigned Languages

Hinglish, Hindi

CANCEL

SAVE CHANGES

4. Promote practice questions submitter to become practice questions reviewer

The practice questions admin can promote question submitters to question reviewers by comparing their stats in the tabular view. The question submitters are sorted based on their performance score.

Performance score: Performance score for question submitter will be based only on the last 100 questions submitted by them. Each accepted question will give one point increase in the performance while each rejected question will decrease performance by two points.

performance score = accepted questions - 2 (rejected questions)

note: only last 100 questions will be considered for this calculation

Edit Role for <username>



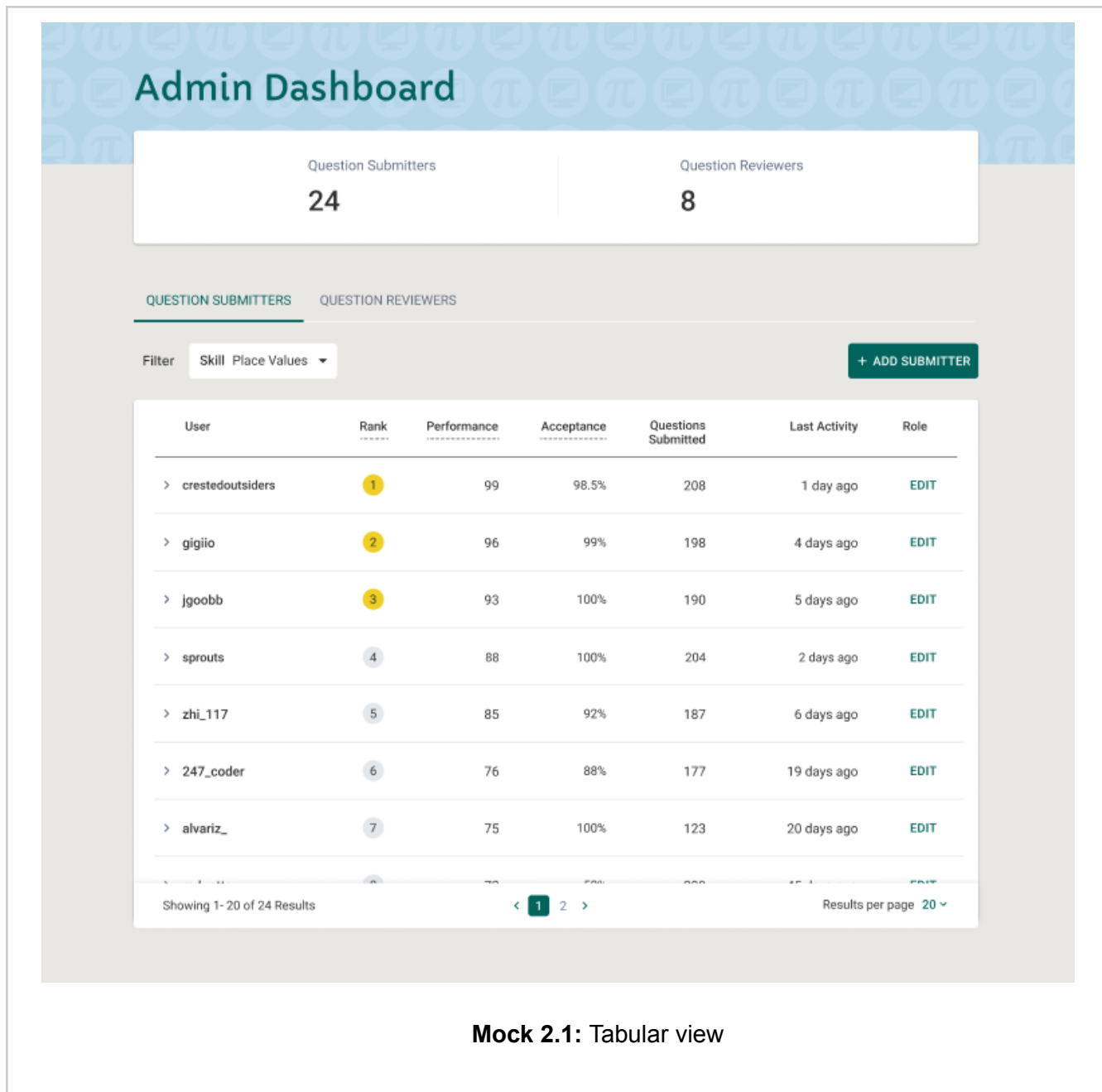
User Role

- ☒ Question Submitter
Submits questions for review
- ☐ Question Reviewer
Reviews questions submitted

CANCEL

SAVE CHANGES

Mock 2.2: Edit role modal



Key High-Level and Architectural Decisions

Decision 1: Model level changes to be decided yet to store the values for evaluating performance metrics

Options to consider for storing data to calculate performance

1) Using existing stats models to store performance score:-

We use the existing stats model to store the performance score. But the risk here is as mentioned by you it will be difficult for us to update all models of the same topic and keeping everything in sync will be problematic.

The model is keyed on (language_code, contributor_user_id, topic_id) for translation and (contributor_user_id, topic_id) for question. This option has conceptual integrity issues. The model is also keyed on topic_id but the data we want to store is not related to topic_id. This will lead to multiple model updates for the same operation.

2) Introduce new storage model to store the performance score-

The mitigation for this problem will be to introduce a new model to store the performance score. This model will have one instance per (contributor_id, language_id) for Translation and per (contributor_id) for Question Coordinator. The performance score will be stored in the form of an array. With each accepted/rejected translation the list will be updated with pop and append operation.

This list will only have three types of values and will have a maximum length of 100.

recent_review_outcomes will look like = [..., "accepted", "rejected", "accepted_with_edits",]

The three allowed values in the list will be "accepted", "rejected" and "accepted_with_edits"

recent_review_outcomes will be unique to each (language_code, contributor_user_id)

We can keep the structure of the model like:-

contributor_id(str)
language_id(str)
recent_review_outcomes(str)
 (accepted/rejected/accepted_with_edits)(Max length of 100)
accepted_translation_count(int)
rejected_translation_count(int)

3) New Storage models with computed value fields:-

We can create a new model with three fields: latest_submitted_translation_count (<= 100), latest_accepted_translation_count, latest_rejected_translation_count. For each update, we will calculate these 3 fields after getting the latest translations of a user with the function `get_user_created_suggestions_by_offset()` in `GeneralSuggestionModel`. This model will

have one instance per (contributor_id, language_id) for Translation and per (contributor_id) for Question Coordinator

Performance measure will be unique to each (contributor_id, language_code) for translation and each (contributor_id) for questions

Factor	Option 1	Option 2	Option 3
Model	Existing(TranslationContributionStatsModel/QuestionContributionStatsModel)	New (TranslationPerformanceStats/QuestionPerformanceStats)	New(TranslationPerformanceStats/QuestionPerformanceStats)
Keyed on	((language_code, contributor_user_id, topic_id)/(contributor_user_id, topic_id))	((language_code, contributor_user_id)/(contributor_user_id))	((language_code, contributor_user_id)/(contributor_user_id))
Duplicate entries	Yes	No	No
Concerned new fields to be added	recent_review_outcomes: list[str]	recent_review_outcomes: list[str] accepted_translation_count(int) rejected_translation_count(int)	latest_submitted_translation_count (<= 100)(int), accepted_translation_count(int), rejected_translation_count(int)
Initial Update	We'll need to calculate performance for each get_multi (contributor_id, language code) and use put_multi to push the initial load. But we have to be careful here cause here same entry might go into multiple models	We'll need to calculate performance for each get_multi (contributor_id, language code) and use put_multi to push the initial load.	We'll need to fetch past 100 models unique to each (contributor_id, language code) get_multi and calculate these three fields: latest_submitted_translation_count, latest_accepted_translation_count, latest_rejected_translation_count. Then put_multi to push the initial data. But here models with pending status will

			also be present which we do not consider while calculating performance
Incremental update With each accepted/rejected translation.	First Get operation to get the existing list. Pop and append operation. put_multi to store all the updated models unique to (contributor_id, language_code)	First Get operation to get the existing list. Pop and append operation. Calculate the two fields again(accepted_translation_count(int) rejected_translation_count(int)). Single put operation to store the updated data.	Get the data about last 100 contribution by get_all called by get_user_created_suggestions_by_offset(). calculate the above mentioned three fields again. Single put operation to store the updated fields. But same here the models with pending status will also be present which we do not consider while calculating performance
Performance For fetching the stats.	GET request for fetching data unique to (contributor_id, language_code).	Single GET request since the model will only have instance per (contributor_id, language_code).	Single GET request to fetch the model unique to (contributor_id, language_code).
Ability to sort	No	Yes	Yes.

According to me option 2 is favorable since

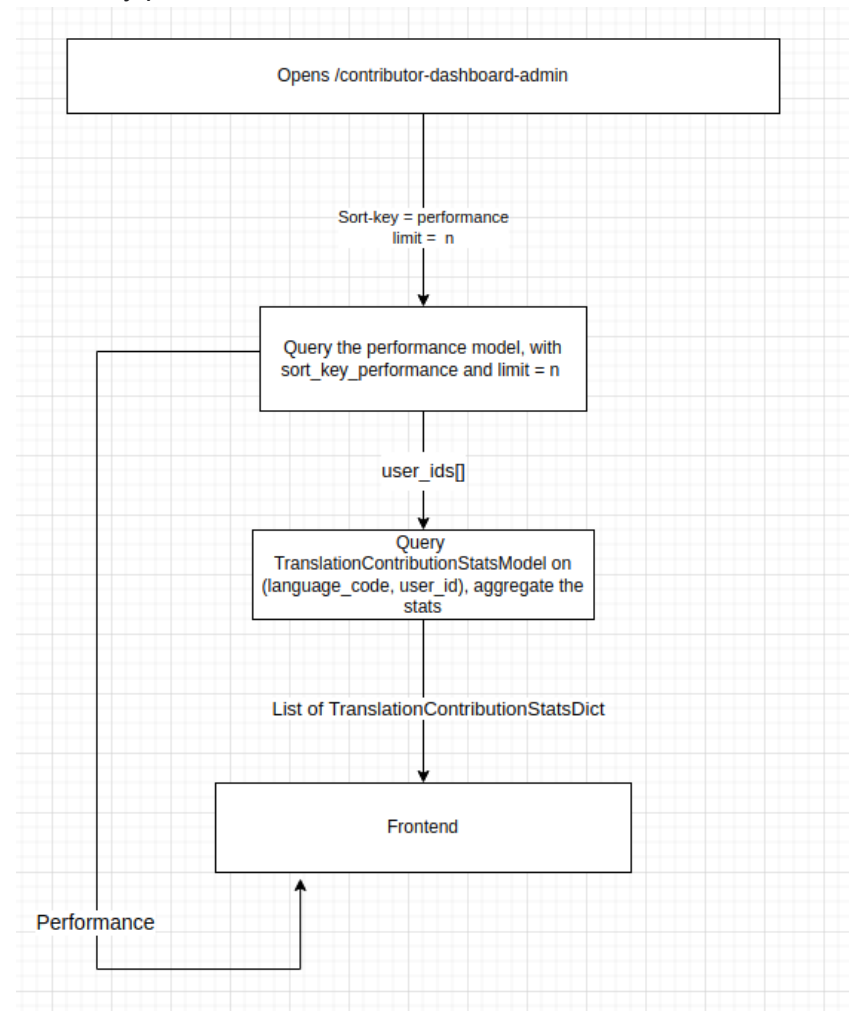
In option 1 we have to update multiple models for single update operation which will be difficult to keep in sync. Also since we won't be storing any numbers sort query won't be possible.

In option 3 we get past 100 suggestions made by the user for a given type which also include suggestions currently in pending_review state. Also while updating the values with each accepted/rejected translation we need to get past 100 models to calculate the values of latest_submitted_translation_count (<= 100)(int), accepted_translation_count(int), rejected_translation_count(int)

According to the discussion and analysis. Option 2 is favorable.

But there's a discrepancy in using this model.

If we use this model to get the stats the flow will be in the following manner.
Sorted by performance and limit = n.



For sort_key=accuracy, we will have a problem in limiting the number of results. Because TranslationContributionStatsModel has one instance per (language_code, contributor_user_id, topic_id) tuple, so to get stats for (language_code, contributor_user_id) we will need to aggregate the stats. Pagination won't be possible in this case.

A solution for this we can modify the new model to store all the required stats at one place which will make pagination and sorting operations fast.

But we can add new fields to this model and update the model with each accepted/rejected translation.

[Chosen Option] New Storage models storing all the stats required for Dashboard

TotalTranslationContributionStatsModel

- Records the translation contribution stats per (contributor_id, language_code) tuple.

- Its IDs will be in the following structure: [contributor_id].[language_code]
- Fields:
 - contributor_id(str). indexed
 - language_code(str). Indexed
 - active_topic_ids(str). indexed
 - recent_review_outcomes(str) indexed
(accepted/rejected/accepted_with_edits)(Max length of 100)
 - recent_performance(int) indexed
 - overall_accuracy(float) indexed
 - last_hundred_accepted_translation_count(int) indexed
 - last_hundred_accepted_translations_without_reviewer_edits_count(int) indexed
 - last_hundred_rejected_translation_count(int) indexed
 - submitted_translations_count: int
 - submitted_translation_word_count: int
 - accepted_translations_count: int
 - accepted_translations_without_reviewer_edits_count: int
 - accepted_translation_word_count: int
 - rejected_translations_count: int
 - rejected_translation_word_count: int
 - last_activity: Set[datetime.date]

Adding extra fields we can get sorted stats with ease.

For Pagination, we can use “**fetch_page()**” function

models, next_cursor, more = query.fetch_page(page_size, start_cursor=cursor)

Args:-

page_size: int. We have to pass this from the frontend. Default=20, Other Possible Values=50, 100

start_cursor:str. The cursor to the next page or None. If None, this means that the search should start from the first page of results.

Returns:-

models: list(SkillSummary). The list of summaries of skills starting at the given cursor.

next_cursor: str or None. A query cursor pointing to the next batch of results. If there are no more results, this might be None.

more: bool. If True, there are (probably) more results after this batch. If False, there are no further results after this batch.

Sorting by each column:-

Performance :-

```
cls.query().order(-cls.recent_performance)
```

Accuracy:-

```
cls.query().order(-cls.accuracy)
```

Last_activity:-

```
cls.query().order(-cls.last_activity)
```

Filtering

By Topic:-

```
cls.query(cls.active_topic_ids.IN(topic_ids))
```

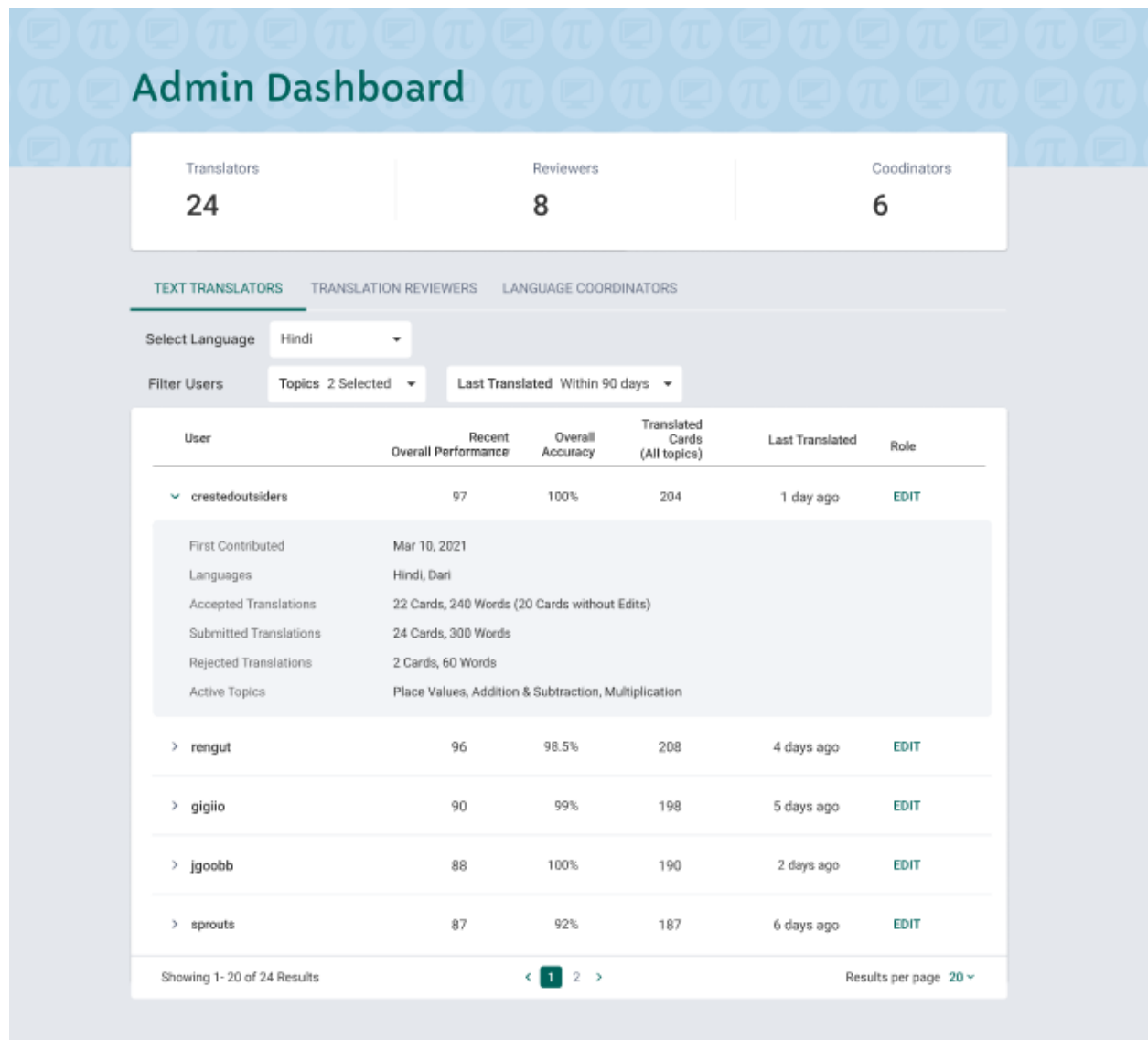
Here *topic_ids* are the topics selected by user in the frontend from the dropdown.

By Last Activity:-

```
cls.query(cls.last_activity > num_days_since_last_activity)
```

Here *num_days_since_last_activity* is the value selected by the user in the frontend.

Options:- none, within 7 days, within 30 days and within 90 days would be options. By default none would be selected.



We can display an info tooltip next to each of the "overall" column names that explains this is for "all" submissions (or reviews, depending on the view) by that contributor.

Decision 2:- Decision table for considering options to store the Language rights of user

[CHOSEN] Option 1:(New LanguageRightsModel)

Introduce a new storage model LanguageRightsModel analogous to TopicRightsModel. This model will be keyed on each language_code. It will have one instance for each language.

Structure of the model:-

language_code(keyed)

coordinators(str)
reviewers(str)
contributors(str)

Option 2:(Existing [UserSettingsModel](#))

Use the existing [UserSettingsModel](#) and append the role of languagecoordinator to the User Roles array, Also add a new array to the model names languages_assigned, all the languages assigned to a coordinator will be appended to this list. The instance of this model is keyed by user_id

Fields to be added in the existing model:-

append role of language_coordinator to the list of user_roles
coordinated_languages(str)

Factor	Option 1	Option 2
Model	New(LanguageRightsModel)	Existing(UserSettingsModel)
Keyed	language_code	user_id
Update the rights	Single get and commit operation to update the rights	Single get and commit operation to update the rights
Getting the stats	Single get() query to get the sequence of models to display stats.	One get() query to get the list of language coordinators and another get() query from <i>CommunityContributionStats Model</i> to get the number of reviewers.
Similarity to existing system	We can follow the same system as TopicRightsModel	This is a new system.

According to the discussion and analysis. Option 1 seems favorable to me.

Option 1 follows the pattern already existing for Topic Rights, we can use the same analogy here. If we consider option 2 we have to design a new system to store the rights of language coordinators along with their assigned languages.

While getting the stats of language coordinators, we'll need to fetch only one model for each language. But in option 2 we need to fetch multiple models for each language.

Decision 3:- Elimination of rank column.

Technical Difficulties:- If we choose to keep the rank column, we can easily rank the users in ascending order. But if we want to rank the users in descending order which is possible only if we have the whole dataset with us. Fetching the whole dataset at once would slow the system.

User Experience:- The stakeholders won't be able to see the rankings of other contributors but still performance and accuracy metrics will be available to the stakeholders to evaluate a contributor. The stakeholders would still be able to sort the contributors on the basis of their performance and accuracy. Stakeholders here are coordinators and web super admin.

Handling Existing data:-

Number of Contributors:-

Currently every user is a contributor but displaying the users with zero contributions in the contribution stats table doesn't make sense. Only contributors with at least one accepted contribution will be considered for displaying their stats in the table.

Existing Questions Admin:-

We are just changing the name of Question Admin to Question Coordinator. The rights to the role remain the same. To do this Web Super Admin needs to follow the below mentioned steps. After we have added the role of Question Coordinator in our system.

Launch Plan:-

- Web Super Admin have to navigate to /admin -> ROLES tab.
- Check for assigned users under the Question Admin role.
- Give user the role of Question Coordinator and at the same time deassign the user to the role of Question Admin.
- After doing this, we can remove the role of Question Admins completely from our system.

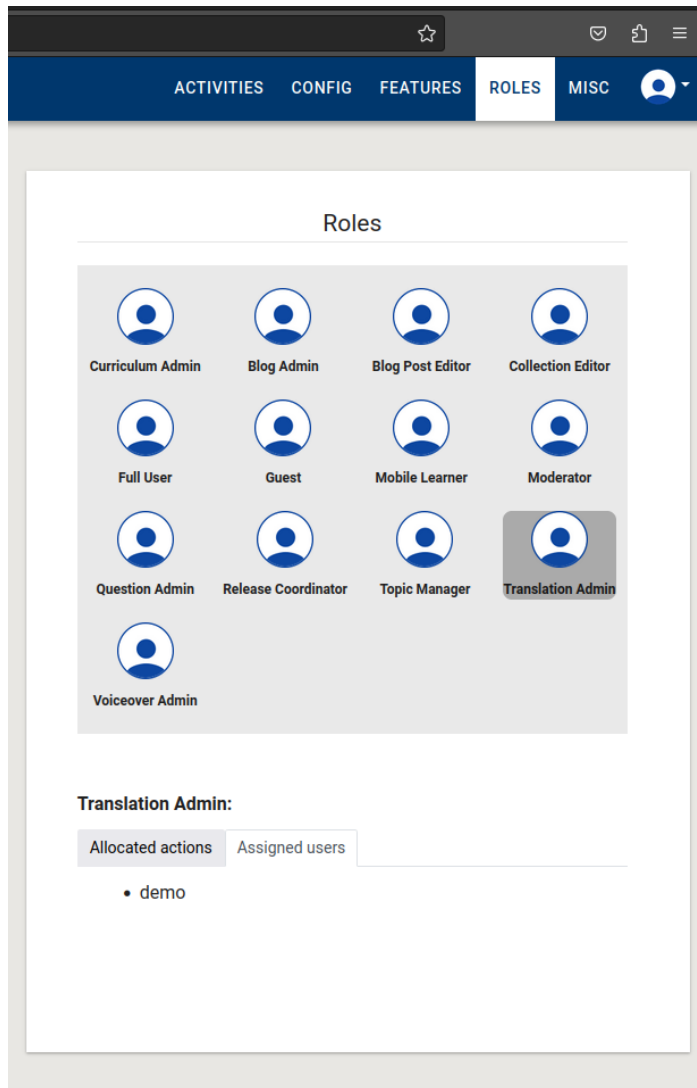
Existing Translation Admin:-

Existing Translation admin don't have any languages assigned to them ie. They can promote contributors from any language to a reviewer. So these admins either need to be promoted to web super admins or assign them a language as a language coordinator. The process for doing this depends on the number of current translation admin. If the numbers are less we can do it manually from the admin dashboard. Once we have a system for adding language coordinators in place, we can follow the below mentioned steps.

Launch Plan:-

- Once the system for promoting a user to language coordinator is ready and deployed. At this point there will be zero language coordinators and N number of translation admins.
- Super admin needs to follow the below mentioned.
- After following the below mentioned steps, we will have N language coordinator and zero translation admins as we will deassign a user as translation admin at the same time we assign him/her as language coordinator.
- Once this is done, we can remove the option to edit the role of translation admin from /admin page -> ROLES tab.

By navigating to /admin, click on the Roles tab. Select Translation Admin as shown below and there we can get the list of all current translation admins.



Steps for promoting existing translation admin (as a web super admin):

- Get the list of current translation admins from `/admin` page, as described above.
- Navigate to `/admin`.
- Enter the user name of the current translation admins, assign the language(s).
- The procedure is similar to adding a new language coordinator.
- After a user is assigned as a language coordinator, deassign the user as translation admin.

Recent suggestion data for calculating performance score:-

// Depending on the storage model decided to store the data.

Implementation Approach

Storage Model Layer Changes

Existing names of Stats model used for user stats.

TranslationContributionStatsModel

TranslationReviewStatsModel

QuestionContributionStatsModel

QuestionReviewStatsModel

Existing stats model are also keyed on topic_id but the stats we need to display doesn't need stats that are keyed on topic_id. Also the new stats models will have data related to last 100 contributions which will be used for calculating performance.

Need to modify the docstring of the existing models as well stating that the difference between both the models.

We are keeping the existing stats models and the new stats model that are added will be named in following pattern.

New Stats Models name:-

TranslationSubmitterTotalContributionStatsModel

TranslationReviewerTotalContributionStatsModel

QuestionSubmitterTotalContributionStatsModel

QuestionReviewerTotalContributionStatsModel

New Storage Models:-

- TranslationSubmitterTotalContributionStatsModel
 - Records the Total Translation contribution stats and data of recent_review keyed per (contributor_id, language_code) tuple. Its IDs will be in the following structure: [language_code] [contributor_id]
 - Fields:
 - contributor_id(str). indexed
 - language_code(str). Indexed
 - topic_ids_with_translation_submissions(str). indexed(repeated)
 - recent_review_outcomes(str) indexed(repeated)

(accepted/rejected/accepted_with_edits)(Max length of 100)

- recent_performance(int) indexed
- overall_accuracy(float) indexed
- submitted_translations_count: int
- submitted_translation_word_count: int

- **accepted_translations_count: int**
 - **accepted_translations_without_reviewer_edits_count: int**
 - **accepted_translation_word_count: int**
 - **rejected_translations_count: int**
 - **rejected_translation_word_count: int**
 - **first_contribution_date: datetime.date**
 - **last_contribution_date: datetime.date**
- **TranslationReviewerTotalContributionStatsModel**
 - Records the Total Translation review stats per (contributor_id, language_code) tuple. Its IDs will be in the following structure:[language_code] [contributor_id]
 - Fields:
 - **contributor_id(str). indexed**
 - **language_code(str). Indexed**
 - **topic_ids_with_translation_reviews(str). indexed(repeated)**
 - **reviewed_translations_count: int. indexed**
 - **accepted_translations_count: int**
 - **accepted_translations_with_reviewer_edits_count: int**
 - **accepted_translation_word_count: int**
 - **rejected_translations_count: int**
 - **first_contribution_date: datetime.date**
 - **last_contribution_date :datetime.date**
- **QuestionSubmitterTotalContributionStatsModel**
 - Records the Total Question Contribution stats per contributor_id.
 - Fields:
 - **contributor_id: str,**
 - **topic_ids_with_question_submissions: str(repeated)**
 - **recent_review_outcomes(str) indexed (repeated)**
(accepted/rejected/accepted_with_edits)(Max length of 100)
 - **recent_performance(int) indexed.**
 - **overall_accuracy(int). Indexed.**
 - **submitted_questions_count: int**
 - **accepted_questions_count: int**
 - **accepted_questions_without_reviewer_edits_count: int**
 - **rejected_questions_count: int**
 - **first_contribution_date: datetime.date**
 - **last_contribution_date: datetime.date**

- QuestionReviewerTotalContributionStatsModel
 - Records the Total Question Review stats per contributor_id.
 - Fields:
 - contributor_id: str,
 - topic_ids_with_question_reviews: str(repeated)
 - reviewed_questions_count: int
 - accepted_questions_count: int
 - accepted_questions_with_reviewer_edits_count: int
 - rejected_questions_count: int
 - first_contribution_date: datetime.date
 - last_contribution_date: datetime.date

Domain Objects

New Domain objects to be introduced:-

TranslationSubmitterTotalContributionStats:

- contributor_id(str). indexed
- language_code(str). Indexed
- topic_ids_with_translation_submissions(str). indexed
- recent_review_outcomes(str) indexed
(accepted/rejected/accepted_with_edits)(Max length of 100)
- recent_performance(int) indexed
- overall_accuracy(float) indexed
- last_hundred_accepted_translations_count(int) indexed
- last_hundred_accepted_translations_without_reviewer_edits_count(int) indexed
- last_hundred_rejected_translations_count(int) indexed
- submitted_translations_count: int
- submitted_translation_word_count: int
- accepted_translations_count: int
- accepted_translations_without_reviewer_edits_count: int
- accepted_translation_word_count: int
- rejected_translations_count: int
- rejected_translation_word_count: int
- first_contribution_date: datetime.date
- last_contribution_date:datetime.date

TranslationReviewerTotalContributionStats:

- contributor_id(str). indexed
- language_code(str). Indexed

- topic_ids_with_translation_reviews(str).
- reviewed_translations_count: int. indexed
- accepted_translations_count: int
- accepted_translations_without_reviewer_edits_count: int
- accepted_translation_word_count: int
- rejected_translations_count: int
- first_contribution_date: datetime.date
- last_contribution_date: datetime.date

QuestionSubmitterTotalContributionStats:

- contributor_id: str,
- topic_ids_with_question_submissions: str
- recent_review_outcomes(str) indexed
(accepted/rejected/accepted_with_edits)(Max length of 100)
- recent_performance(int) indexed.
- overall_accuracy(float). Indexed.
- last_hundred_submitted_questions_count: int
- last_hundred_accepted_questions_count: int
- last_hundred_accepted_questions_without_reviewer_edits_count: int
- submitted_questions_count: int
- accepted_questions_count: int
- accepted_questions_without_reviewer_edits_count: int
- rejected_questions_count: int
- first_contribution_date: datetime.date,
- last_contribution_date: datetime.date

QuestionReviewerTotalContributionStats:

- contributor_id: str
- topic_ids_with_question_reviews: str
- reviewed_questions_count: int
- accepted_questions_count: int
- accepted_questions_without_reviewer_edits_count: int
- rejected_questions_count: int.
- first_contribution_date: datetime.date,
- last_contribution_date: datetime.date

User Flows (Controllers and Services)

Services :

All the services will be in the following file core/domain/contribution_stats_services.py

Services to be added for getting stats :-

- get_translation_submitter_total_stats_from_model

"""Returns a domain object for TranslationSubmitterTotalContributionStats model.

Args:

translation_submitter_model:

TranslationSubmitterTotalContributionStatsModel. Model to get corresponding domain object.

Returns:

TranslationSubmitterTotalContributionStats. The domain object corresponding to given model.

"""

- get_translation_reviewer_total_stats_from_model

"""Returns a domain object for TranslationReviewerTotalContributionStats model.

Args:

translation_reviewer_model:

TranslationReviewerTotalContributionStatsModel. Model to get corresponding domain object.

Returns:

TranslationreviewerTotalContributionStats. The domain object corresponding to given model.

"""

- get_question_submitter_total_stats_from_model

"""Returns a domain object for QuestionSubmitterTotalContributionStats model.

Args:

question_submitter_model:

QuestionSubmitterTotalContributionStatsModel. Model to get corresponding domain object.

Returns:

QuestionSubmitterTotalContributionStats. The domain object corresponding to given model.

"""

- get_question_reviewer_total_stats_from_model

"""Returns a domain object for QuestionReviewerTotalContributionStats model.

Args:

question_reviewer_model:

QuestionReviewerTotalContributionStatsModel. Model
to get corresponding domain object.

Returns:

QuestionreviewerTotalContributionStats. The domain object
corresponding to given model.

"""

-
- get_translation_submitter_total_stats

"""Returns the list of domain objects according to values specified.

Args:

page_size: int. Number of models to fetch.

offset: int. Number of results to skip from the beginning of all
results matching the query.

language_code: str. The language code to get results for.

sort_by: SortChoices|None. A string indicating how to sort the
result.

topic_ids: List[str]|None. List of topic ID(s) to fetch
contributor stats for.

num_days_since_last_activity: int. To get number of users
who are active in num_days_since_last_activity.

Returns:

3-tuple(sorted_results, next_offset, more). where:

sorted_results:

list(TranslationSubmitterTotalContributionStats).

The list of domain objects which match the supplied
language_code, topic_ids and num_days_since_last_activity
filters, returned in the order specified by sort_by.

next_offset: int. Number of results to skip in next batch.

more: bool. If True, there are (probably) more results after
this batch. If False, there are no further results
after this batch.

"""

- get_translation_reviewer_total_stats

"""Returns the list of domain objects according to values specified.

Args:

page_size: int. Number of models to fetch.

offset: int. Number of results to skip from the beginning of all results matching the query.
language_code: str. The language code to get results for.
sort_by: SortChoices|None. A string indicating how to sort the result.
num_days_since_last_activity: int|None. To get number of users who are active in num_days_since_last_activity.

Returns:

3-tuple(sorted_results, next_offset, more). where:
sorted_results:
list(TranslationReviewerTotalContributionStats).
The list of domain objects which match the supplied language_code, and num_days_since_last_activity filters, returned in the order specified by sort_by.
next_offset: int. Number of results to skip in next batch.
more: bool. If True, there are (probably) more results after this batch. If False, there are no further results after this batch.

"""

- get_question_submitter_total_stats

"""Returns the list of domain objects according to values specified.

Args:

page_size: int. Number of results to fetch.
offset: int. Number of results to skip from the beginning of all results matching the query.
sort_by: SortChoices|None. A string indicating how to sort the result.
topic_ids: List[str]|None. List of topic ID(s) to fetch contributor stats for.
num_days_since_last_activity: int. To get results of users who are active in num_days_since_last_activity.

Returns:

3-tuple(sorted_results, next_offset, more). where:
sorted_results:
list(QuestionSubmitterTotalContributionStats).
The list of domain objects which match the supplied topic_ids and num_days_since_last_activity filters, returned in the order specified by sort_by.
next_offset: int. Number of results to skip in next batch.
more: bool. If True, there are (probably) more results after

this batch. If False, there are no further results after this batch.

"""

- `get_question_reviewer_total_stats`

"""Returns the list of domain objects according to values specified.

Args:

`page_size`: int. Number of results to fetch.

`offset`: int. Number of results to skip from the beginning of all results matching the query.

`sort_by`: SortChoices|None. A string indicating how to sort the result.

`num_days_since_last_activity`: int|None. To get result of users who are active in `num_days_since_last_activity`.

Returns:

3-tuple(sorted_results, next_offset, more). where:

`sorted_results`:

`list(QuestionReviewerTotalContributionStats)`.

The list of domain objects which match the supplied `num_days_since_last_activity` filter, returned in the order specified by `sort_by`.

`next_offset`: int. Number of results to skip in next batch.

`more`: bool. If True, there are (probably) more results after this batch. If False, there are no further results after this batch.

"""

Services to be used for updating the new stats models:-

[update_translation_contribution_stats_at_submission.](#)

[update_translation_contribution_stats_at_review](#)

[update_translation_review_stats](#)

[update_question_contribution_stats_at_submission](#)

[update_question_contribution_stats_at_review](#)

[Update_question_review_stats](#)

These services currently update the stats model used to display contributor stats(`TranslationContributionStatsModel`, `TranslationReviewStatsModel`,).

These services will be modified to update new models used for this project as well(TranslationSubmitterTotalContributionStatsModel, TranslationReviewerTotalContributionStatsModel,).

get_language_coordinator_stats :

Args:

None.

Returns:

list(**LanguageRightsModel**). LanguageCoordinatorStats domain objects.

update_recent_translation_performance_stat():

"""Get RecentTranslationPerformanceStatsModel based on user_id and language_code, update the recent_review_outcomes, calculate accepted_translation_count, rejected_translation_count and performance score, then store them into RecentTranslationPerformanceStatsModel.

Args:

User_id:

Language_code:

Review_stat: str ('accepted'/'rejected')

Returns:

None

"""

It will use given services to get the number of accepted and rejected contributions and calculate the score based on:

$$\text{performance score} = \text{accepted questions} - 2 (\text{rejected questions})$$

(only last 100 contributions will be considered for this calculation)

Controllers:

Existing Controllers:

[ContributorStatsSummariesHandler](#) :

- Existing Endpoint to fetch contributor stats for supplied contribution type and user id.

[ContributorUsersListHandler](#) :

- Existing Endpoint to list all the user with supplied category and language code

[ContributionRightsHandler](#) :

- Existing Endpoint to edit user rights given contribution category, user_id

[ContributionRightsDataHandler](#) :

- Existing Endpoint to show contributor rights of a user.

The above mentioned endpoints will not be changed but rather be used as it is.

AllContributorStatsSummariesHandler:

- New Endpoint to fetch all the contributor stats given contribution type.

Fetches contribution stats of category and sub_category

Args:

contribution_type: Type of contribution(translation/question)
contribution_subtype: Sub type of contribution (contribution/review)
ITEMS_PER_PAGE: int. Maximum number of entities to be returned. Default: 20. Other possible options: 50, 100.
language_code: str. The language to retrieve stats for, if contribution_type=translation.
offset: int. The number of results to skip from the beginning of all results matching the query.
sort_key: str|None. The key to sort the suggestions by.

Returns: The contributor stats for the given category, based on the given inputs. For example, if the input supplied is (contribution_type, contribution_subtype) = ("translation", "contribution"), the handler will return a sequence of TranslationContributionStats domain objects.

```
ListContributorDashboardStatsTypes = Sequence[Union[
    suggestion_registry.TranslationContributionStats,
    suggestion_registry.TranslationReviewStats,
    suggestion_registry.QuestionContributionStats,
    suggestion_registry.QuestionReviewStats
]]
```

"""

LanguageCoordinatorRightsHandler:-

We can use the existing handler ([AdminRoleHandler](#)) to update the role of question coordinator but for language coordinator we need to assign a particular language(analogous to how we do

for topic manager). Currently we have a different handler ([TopicManagerRoleHandler](#)) for managing topic related rights.

This handler will be used to give a user, or withdraw from a user, the language coordinator role for a particular language.

And param to be passed is

```
username: {
  'schema': {
    'type': 'basestring'
  }
},
'action': {
  'schema': {
    'type': 'basestring',
    'choices': ['add', 'remove']
  }
},
'language_code': {
  'schema': {
    'type': 'basestring'
  }
}
```

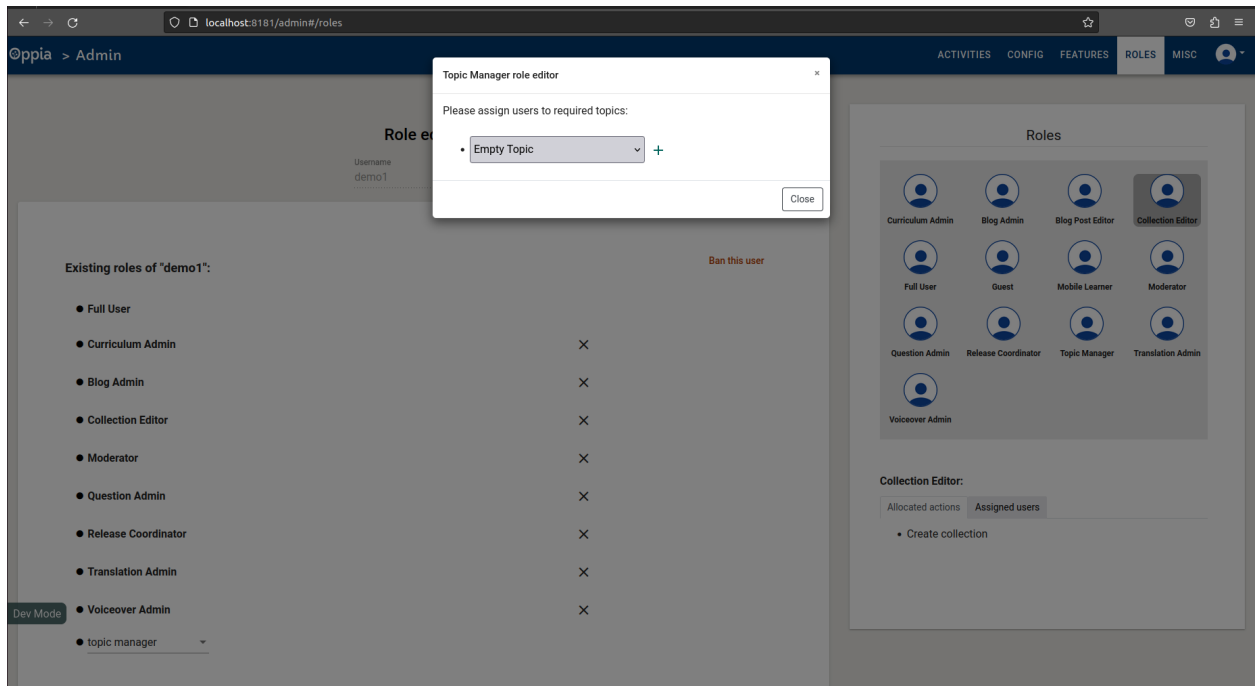
Web frontend changes

A full revamp to

core/templates/pages/contributor-dashboard-admin-page/contributor-dashboard-admin-page.component.html

Existing component to be used:-

1. We can use the existing modal (snapshot added below) of adding a topic manager. But instead of filling the dropdown with topics we will populate the dropdown with [SUPPORTED AUDIO LANGUAGES](#), we are going to rename it as **is_supported_translation_language_code** as discussed previously.



New Components to add :

core/templates/pages/contributor-dashboard-admin-page/tabs/

- core/templates/pages/contributor-dashboard-admin-page/tabs/text-translators
 - Displays the stats of all text translators in tabular format [link](#)

Admin Dashboard

Translators

24

Reviewers

8

Coordinators

6

TEXT TRANSLATORS

TRANSLATION REVIEWERS

Filter

Language Hindi

Topics 2 Selected

Last Translated Within 90 days

+ ADD TRANSLATOR

User	Rank	Performance	Accuracy	Translated Cards	Last Translated	Role
crestedoutsiders	1	97	100%	204	1 day ago	EDIT
<div> <div>Date Joined</div> <div>Mar 10, 2021</div> </div> <div> <div>Languages</div> <div>Hindi, Dari</div> </div> <div> <div>Accepted Translations</div> <div>22 Cards, 240 Words (20 Cards without Edits)</div> </div> <div> <div>Submitted Translations</div> <div>24 Cards, 300 Words</div> </div> <div> <div>Rejected Translations</div> <div>2 Cards, 60 Words</div> </div> <div> <div>Active Topics</div> <div>Place Values, Addition & Subtraction, Multiplication</div> </div>						
rengut	2	96	98.5%	208	4 days ago	EDIT
giglio	3	90	99%	198	5 days ago	EDIT
jgoobb	4	88	100%	190	2 days ago	EDIT
sprouts	5	87	92%	187	6 days ago	EDIT

Showing 1- 20 of 24 Results

1

2

Results per page: 20

- core/templates/pages/contributor-dashboard-admin-page/tabs/translation-reviewers
 - Displays stats of all text reviewers [link](#)

Admin Dashboard

Translators
24

Reviewers
8

Coordinators
6

TEXT TRANSLATORS

TRANSLATION REVIEWERS

Filter

Languages English, Hindi

Last Reviewed Within 90 days

User	Reviewed Cards	Last Reviewed	Role
<div>partner</div> <div> <div>Date Joined</div> <div>Mar 10, 2021</div> </div> <div> <div>Languages</div> <div>Hindi</div> </div> <div> <div>Accepted Translations</div> <div>22 Cards, 240 Words (20 Cards without Edits)</div> </div> <div> <div>Rejected Translations</div> <div>24 Cards, 300 Words</div> </div> <div> <div>Active Topics</div> <div>Place Values, Addition & Subtraction, Multiplication</div> </div>	64	1 day ago	EDIT
> momo0425	58	4 days ago	EDIT
> fumya	54	3 days ago	EDIT
> antowatasi	49	12 days ago	EDIT
> nagano0917	48	20 days ago	EDIT

Showing 1- 20 of 24 Results

< 1 2 >

Results per page 20

- core/templates/pages/contributor-dashboard-admin-page/tabs/question-submitter
 - Displays stats of all question submitters: [link](#)

Filter Skill Place Values Last Submitted Within 90 days + ADD SUBMITTER

User	Rank	Performance	Acceptance	Questions Submitted	Last Submitted	Role
> crestedoutsiders	1	99	98.5%	208	1 day ago	EDIT
<div>Date JoinedMar 10, 2021</div> <div>Submitted Questions24</div> <div>Accepted Questions22</div> <div>Accepted (No Edits)22</div>						
Active Skills		Submitted Questions	Accepted Questions	Accepted (No Edits)		
Place Values		12	12	12		
Addition and Subtraction		8	6	5		
Multiplication		4	4	5		
> giglio	2	96	99%	198	4 days ago	EDIT
> jgoobb	3	93	100%	190	5 days ago	EDIT

Showing 1 - 20 of 24 Results < 1 2 > Results per page 20

4. core/templates/pages/contributor-dashboard-admin-page/tabs/question-reviewer
 - Displays all the stats of question reviewers [link](#)

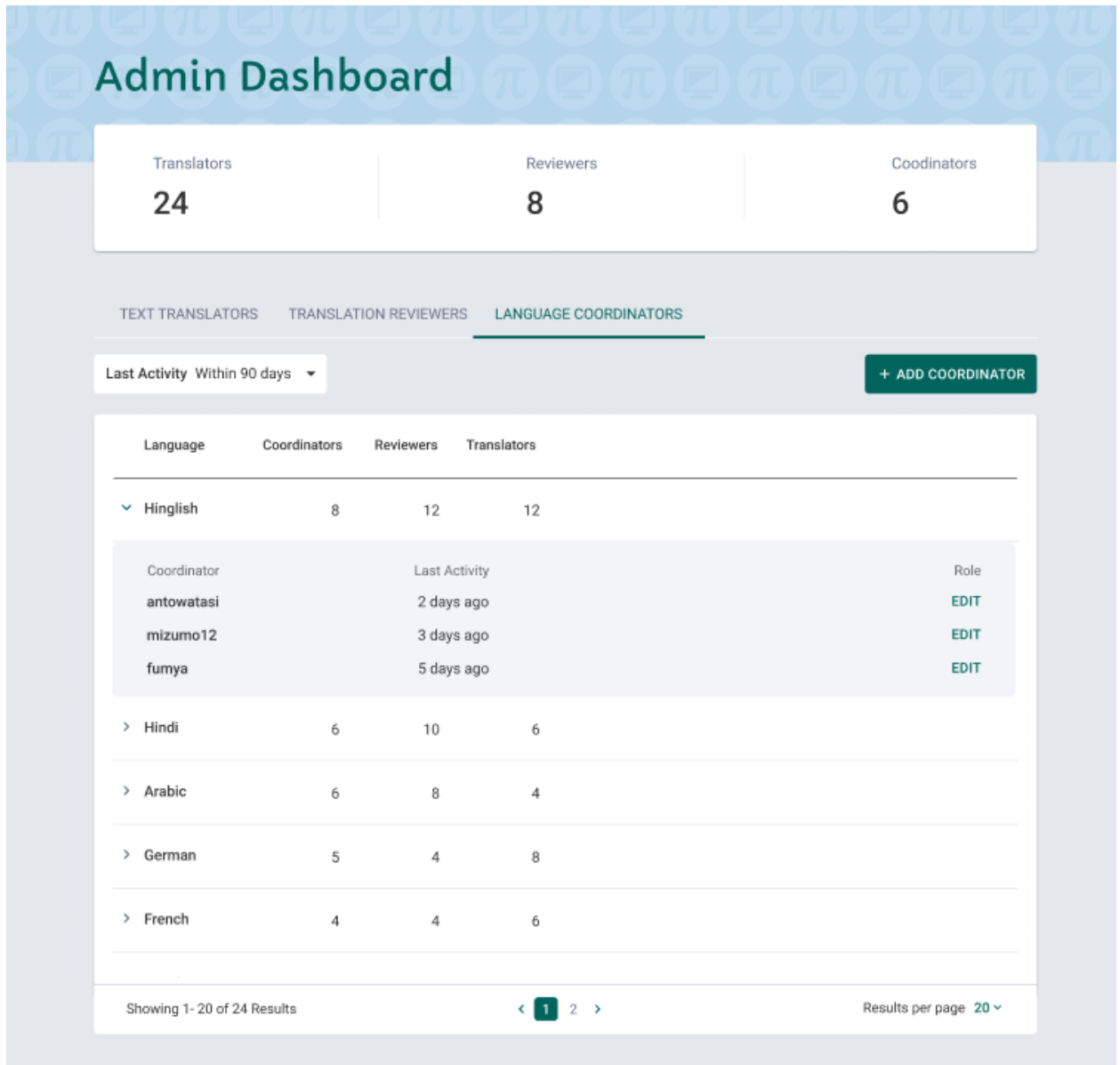
QUESTION SUBMITTERS QUESTION REVIEWERS

Filter Skill Place Values Last Reviewed Within 90 days

User	Questions Reviewed	Last Reviewed	Role
> brokenupon	75	2 days ago	EDIT
<div>Date JoinedMar 10, 2021</div> <div>Reviewed Questions24</div> <div>Accepted Questions22</div>			
Active Skills		Reviewed Questions	Accepted Questions
Place Values		12	12
Addition and Subtraction		8	5
Multiplication		4	5
> backacheyearling	54	1 day ago	EDIT
> nooksseverance	48	4 days ago	EDIT

Showing 1 - 20 of 24 Results < 1 2 > Results per page 20

5. core/templates/pages/contributor-dashboard-admin-page/tabs/language-coordinator
 - Displays all the stats of language coordinators [link](#)

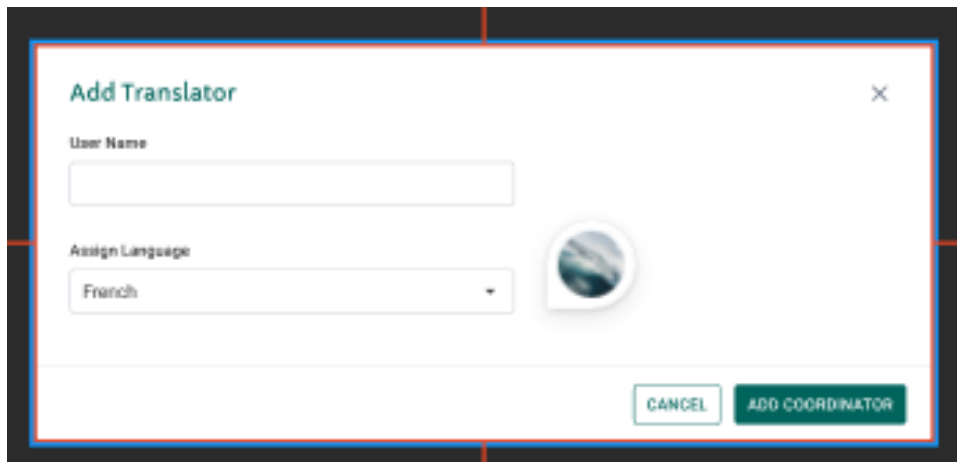


6. `core/templates/pages/contributor-dashboard-admin-page/tabs/question-coordinator`
 For the language coordinators we display the stats for particular languages but this is not the case for question coordinators. For Question coordinators, I think a simple list will do the work.

Question Coordinator	Last Activity
Coordinator 1	1 day ago
Coordinator 2	3 days ago
.....
...

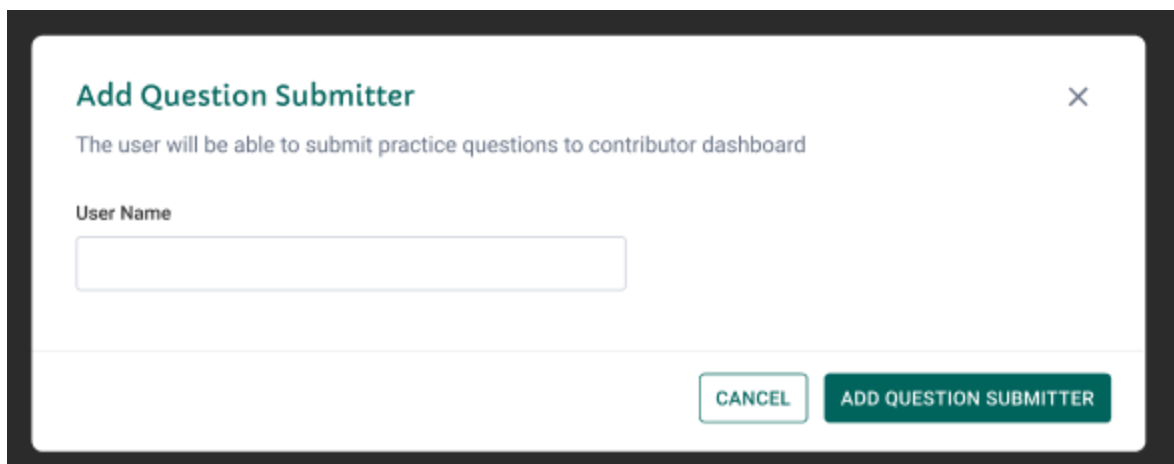
core/templates/pages/contributor-dashboard-admin-page/modals/

2. core/templates/pages/contributor-dashboard-admin-page/modals/add-translator
- Modal to add translator. [Link](#)



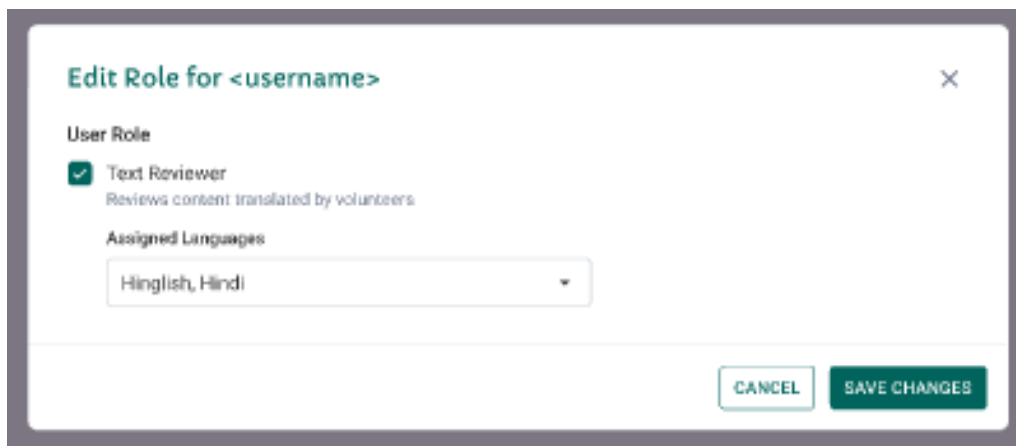
The 'Add Translator' modal is a white box with a red border and a close button (X) in the top right corner. It contains a 'User Name' text input field, an 'Assign Language' dropdown menu with 'French' selected, a circular profile picture placeholder, and two buttons at the bottom: 'CANCEL' and 'ADD COORDINATOR'.

3. core/templates/pages/contributor-dashboard-admin-page/modals/add-question-submitter
- Modal to add question submitters. [Link](#)



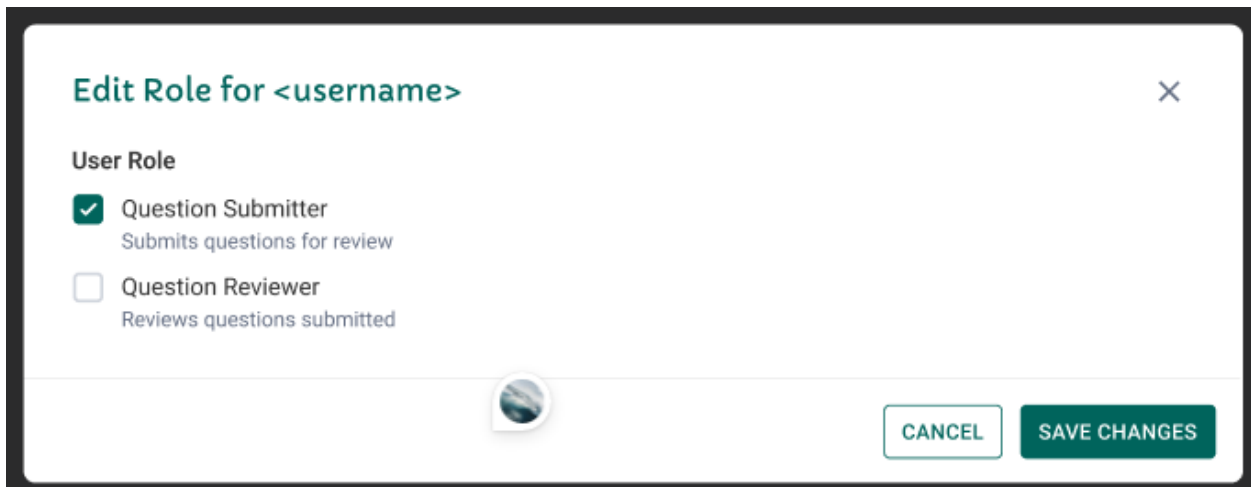
The 'Add Question Submitter' modal is a white box with a close button (X) in the top right corner. It features a title 'Add Question Submitter', a description 'The user will be able to submit practice questions to contributor dashboard', a 'User Name' text input field, and two buttons at the bottom: 'CANCEL' and 'ADD QUESTION SUBMITTER'.

4. core/templates/pages/contributor-dashboard-admin-page/modals/edit_role_translations
- Modal to edit role of a contributor in translation category [Link](#)



The 'Edit Role for <username>' modal is a white box with a close button (X) in the top right corner. It contains a 'User Role' section with a checked checkbox for 'Text Reviewer' and a description 'Reviews content translated by volunteers'. Below this is an 'Assigned Languages' dropdown menu with 'Hinglish, Hindi' selected. At the bottom are two buttons: 'CANCEL' and 'SAVE CHANGES'.

5. core/templates/pages/contributor-dashboard-admin-page/modals/edit-role-questions
- Modal to edit the role of a contributor in the question category. [Link](#)




Edit Role for <username> ×

User Role

☒ **Question Submitter**
Submits questions for review

☐ **Question Reviewer**
Reviews questions submitted



CANCEL **SAVE CHANGES**

Testing Plan

Initial Setup:-

LANGUAGE COORDINATOR CUJ

- Create a user with testadmin@example.com(admin account) and username admin
- Create a dummy user with dummy@example.com and username dummy
- Create Opportunities
 - Create a topic with name “abc”.
 - Save and publish the topic.
 - Create exploration1 and exploration2, with one card in each exploration.
 - Save and publish the exploration.
 - Go to /topics-and-skills-dashboard.
 - add exploration1 as chapter to abc, exploration2 as chapter to abc.
- Create 5 different users and assign them following roles
 - Translation contributor
(email:- translationcontributor1@example.com
username:- translationcontributor1)
 - Translation contributor
(email:- translationcontributor2@example.com
username:- translationcontributor2)
 - Translation reviewer(Review rights for hindi language)
(email:- translationreviewer1@example.com
username:- translationreviewer1)

- Translation reviewer(Review rights for hindi language)
(email:- translationreviewer2@example.com
username:- translationreviewer2)
- Language Coordinator
(email:- languagecoordinator@example.com username:- languagecoordinator)
- Login as translationcontributor1
 - navigate to /contributor-dashboard, submit a translation for exploration1 for hindi language.
- Logout as translationcontributor1
- Login as translationreviewer1
 - navigate to /contributor-dashboard, click on review translation and review and accept the translation for exploration1 submitted by translationcontributor1
- Logout as translationreviewer1
- Login as translationcontributor2
 - navigate to /contributor-dashboard, submit a translation for exploration2 for hinglish language.
- Logout as translationcontributor2
- Login as translationreviewer2
 - navigate to /contributor-dashboard, click on review translation and review and reject the translation for exploration2 submitted by translationcontributor2
- Logout as translationreviewer2
- Login in as testadmin@example.com
 - Go to /admin and assign languagecoordinator with the role of Language Coordinator for hindi and hinglish.

#	Test name	Initial setup step	Step	Expectation
1.	Language Coordinator Tests	Login as languagecoordinator@example.com	Navigate to contributor-dashboard-admin	Page should load and Table with 2 tabs should be visible Mock 1) Translation Contributors 2) Translation Reviewers Tabs that won't be visible. 1) Language Coordinators 2) Question Submitters. 3) Question Reviewers. 4) Question Coordinators.
			Clicking on a Text Translator(Though it will be defaulted when the page loads)	A table with 7 columns should load Mock . The table will be having 2 rows with following values:- Row1:- 1) User:- translationcontributor1 2) Performance:- 1 3) Accuracy:- 100 4) Translated Cards:- 1 5) Last Translated:- today 6) Role (With Edit option)

			Clicking on the translationcontributor1 i.e. selecting a row	Expanded view of translationcontributor1 stats should be visible consisting of <ol style="list-style-type: none"> 1) Date Joined:- Month DD, YYYY 2) Languages:- Hindi 3) Accepted Cards:- 1 4) Submitted Cards:- 1 5) Rejected Cards:- 0 6) Active Topics:- abc
			Toggling the EDIT button on translationcontribution1 row	Edit role dialog box should open and Language coordinator should be able to update the roles of a contributor. The modal will have the title "Edit role for translationcontribution1" Translation Reviewer checkbox unticked Assigned Languages dropdown will be grayed out.
			Select Hinglish from language dropdown	A table with 7 columns should load Mock . The table will be having 2 rows with following values:- Row1:- <ol style="list-style-type: none"> 1) User:- translationcontributor2 2) Performance:- -2 3) Accuracy:- 0 4) Translated Cards:- 1 5) Last Translated:- today 6) Role (With Edit option)
			Navigate to Text Reviewers	Table with 4 columns should load with following columns Mock . The table will be having 2 rows: Row1:- <ol style="list-style-type: none"> 1) User:- translationreviewer1 2) Reviewed Cards:- 1 3) Last Reviewed:- today 4) Role:- EDIT Row2:- <ol style="list-style-type: none"> 1) User:- translationreviewer2 2) Reviewed Cards:- 1 3) Last Reviewed:- today 4) Role:- EDIT
			Clicking on EDIT button in role column of translationreviewer1	An Edit Role Modal must open with title "Edit Role for translationreviewer1" Translation Reviewer must be ticked. Language(s) selected in dropdown must be Hindi.
			Click on a translationreviewer1 i.e. selecting a row	Expanded view of user stats should load consisting of following information <ol style="list-style-type: none"> 1) Date Joined:- Month dd, YYYY 2) Languages:- Hindi

				3) Accepted Cards:- 1 4) Rejected Cards:- 0 5) Active Topics:- abc
--	--	--	--	--

QUESTION COORDINATOR CUJ

- Create a user with testadmin@example.com(admin account) and username admin
- Create a dummy user with dummy@example.com and username dummy
- Create Opportunities
 - Create a topic with name “**abc**”.
 - Save and publish the topic.
 - Create **exploration1** and **exploration2**, with one card in each exploration.
 - Save and publish the exploration.
 - Go to /topics-and-skills-dashboard.
 - add exploration1 as chapter to abc, exploration2 as chapter to abc.
- Create **Skill1** and assign it to topic “abc”
- Create 5 different users and assign them following roles
 - Question Submitter
(email:- questionsubmitter1@example.com
username:- questionsubmitter1)
 - Question Submitter
(email:- questionsubmitter2@example.com
username:- questionsubmitter2)
 - Question reviewer(Question review rights)
(email:- questionreviewer1@example.com
username:- questionreviewer1)
 - Question reviewer(Question review rights)
(email:- questionreviewer2@example.com
username:- questionreviewer2)
 - Question Coordinator
(email:- questioncoordinator@example.com username:- questioncoordinator)
- Login as questionsubmitter1
 - navigate to /contributor-dashboard, submit a question for exploration1.
 - Submitting a question:-
 - Goto Contributor dashboard, click on Submit question tab, click on Suggest question button next to 'exploration1'
 - - Select difficulty level as Medium and click on Done, Enter ""Test Question"" in Type your question here and click on Done
 - - Click on Add Interaction, select Text Input and in Customize interaction dialog box click on Save interaction button
 - - In Add Response dialog box select rule 'contains at least one of', in element enter 'a', enter 'correct' in Oppia tells the learner and check the checkbox 'The answers in this group are correct' and click on Save and

- Add another click on Tag with misconception, select the misconception and click on Done. Click on Save Response button
- - Click on default response, enter 'incorrect' in Oppia tells the learner and Save feedback
 - - Click on Add Hint button, Enter 'Any word with letter a'. Click on Save Hint button
 - - Click on Add Solution button, select 'One Solution' is and enter 'and', in the explanation 'enter any word with a'. Click on Check and Save solution button
 - - Click on Submit question button
- Logout as questionsubmitter1
 - Login as questionreviewer1
 - navigate to /contributor-dashboard, click on review questions and review and accept the question for exploration1 submitted by questionsubmitter1
 - Logout as translationreviewer1
 - Login as questionsubmitter2
 - navigate to /contributor-dashboard, submit a question for exploration2.
 - Process for submitting a question is same as above
 - Logout as questionsubmitter2
 - Login as questionreviewer2
 - navigate to /contributor-dashboard, click on review question and review and reject the question for exploration2 submitted by questionsubmitter2
 - Logout as questionreviewer2
 - Login in as testadmin@example.com
 - Go to /admin and assign questioncoordinator with the role of Question Coordinator.

2.	Question Coordinator tests	Login as Question Coordinator	Navigating to contributor-dashboard-admin page	Page should load Table with 2 tabs should be visible Mock 1) Question Submitters 2) Question Reviewers Tabs that won't be visible 1) Text translators 2) Translation Reviewers 3) Language Coordinators 4) Question Coordinator
			Clicking on Question Submitters (Though it will be selected as default option when the page loads)	Table with below mentioned 7 columns should load Mock . The table will show 2 rows with following data column wise Row 1:- 1) User:- questionsubmitter1 2) Performance:-1 3) Acceptance:- 100 4) Questions Submitted:- 1 5) Last Submitted:- today 6) Role:- EDIT

				<p>Row 2:-</p> <ol style="list-style-type: none"> 1) User:- questionsubmitter2 2) Rank:- 2 3) Performance:- -2 4) Acceptance:- 0 5) Questions Submitted:- 1 6) Last Submitted:- today 7) Role:- EDIT
			Selecting questionsubmitter1 i.e. selecting a row	<p>Expanded view with user stats should be visible consisting of</p> <ol style="list-style-type: none"> 1) Date Joined:- Month dd, YYYY 2) Accepted Questions:- 1 3) Accepted Questions (No edits):- 1 4) Table with bifurcation of question within different skills <ol style="list-style-type: none"> a) Skill1:- <ul style="list-style-type: none"> - Submitted Questions:- 1 - Accepted Questions:- 1 - Accepted Questions(no edits):- 1
			Toggling Add Submitter button	<p>Modal to add question submitter should open.</p> <p>Mock.</p>
			Enter the username "dummy" in the text field of the modal and click on ADD QUESTION SUBMITTER	<p>dummy should be displayed as a new row in the Question Submitter stats table.</p>
			Navigating to Question Reviewers tab	<p>Table with below mentioned 4 columns and 2 rows should load Mock</p> <p>Row 1:-</p> <ol style="list-style-type: none"> 1) User: questionreviewer1 2) Questions Reviewed: 1 3) Last Reviewed: today 4) Role: EDIT <p>Row 2:-</p> <ol style="list-style-type: none"> 1) User:- questionreviewer2 2) Questions Reviewed: 1 3) Last Reviewed: today 4) Role: EDIT
			Selecting questionreviewer1 i.e. selecting a row.	<p>Expanded view with user stats should be visible consisting of</p> <ol style="list-style-type: none"> 1) Date Joined:- Month dd, YYYY 2) Reviewed Questions: 1 3) Accepted Questions: 1 4) Table with bifurcation of questions reviewed and accepted in different skills <p>Row 1 in expanded view will consist of the following column wise data</p> <ul style="list-style-type: none"> - Active Skill:- Skill1

				<ul style="list-style-type: none"> - Reviewed Questions: 1 - Accepted Questions: 1
			Toggling EDIT button under role column of questionreviewer1	Modal for editing the role of Question Reviewer should appear. Mock . The modal will have two checkboxes:- Question Submitter - unticked Question Reviewer - ticked
			Untick the checkbox of the question reviewer and save changes.	questionreviewer1 should disappear from the list of question reviewers.

Web Super Admin CUJ :-

- Create a user superadmin@gmail.com and username admin
- Create two users languagecoordinator@gmail.com and questioncoordinator@gmail.com with usernames languagecoordinator and questioncoordinator respectively.

1	Login as superadmin@gmail.com	Navigate to /admin -> ROLES tab	
		Enter the username as languagecoordinator. Toggle edit button.	
		Click on Add role	Language Coordinator role should be present in the dropdown.
		Select Language Coordinator	Modal to select the topic pops up.
		Select language as Hindi from dropdown.	Hindi appears in a bullet list in the same model.
		Close the model	Language Coordinator appears in the list of roles of the user languagecoordinator.
		Enter the username as questioncoordinator in the input field	

		and toggle edit icon.	
		Click on add role	Question Coordinator role is present in the dropdown
		Select Question Coordinator from the dropdown	Question Coordinator is added to the list of roles of the user questioncoordinator.
		Navigate to /contributor-dashbo ard-admin -> Language Coordinators tab	1 row appears in the following format Language:- Hindi Coordinators:- 1 Reviewers:- 0 Translators: 0
		Clicking on Hindi row	An expanded view of the language will be displayed with the following contents Coordinator:- languagecoordinator Last Activity:- today Role: EDIT

Implementation Plan

Plan for Populating stats using beam jobs.

New job file in core/jobs/batch_jobs/contributor_admin_stats_computation_jobs.py.

Required Models:-

We need to populate four new models:-

TranslationSubmitterTotalContributionStatsModel,
TranslationReviewerTotalContributionStatsModel,
QuestionSubmitterTotalContributionStatsModel,
QuestionReviewerTotalContributionStatsModel

For populating fields in these models, we need two different models.

For fields that help in calculating performance

(

recent_review_outcomes

) we need GeneralSuggestionsModel

and for the rest of the fields we need respective stats models, for example for TranslationSubmitterTotalContributionStatsModel we need TranslationContributionStatsModel.

Getting the required data

class GenerateContributionAdminStats

translation_submission_stats_models :- Pcollection of [TranslationContributionStatsModel](#) (Grouped by id(language_code, contributor_id))

last_100_general_suggestions_translation_model:- Pcollection of [GeneralSuggestionModel](#).
Filtered for (suggestion_type= SUGGESTION_TYPE_TRANSLATE_CONTENT AND status!=STATUS_IN_REVIEW.
Sort in created_on order with limit Of 100)

translation_reviewer_stats_models:- Pcollection of [TranslationReviewStatsModel](#) (Grouped by id(language_code, contributor_id))

question_submission_stats_models:- Pcollection of [QuestionContributionStatsModel](#) (Grouped by id(contributor_id))

last_100_general_suggestions_question_model:- Pcollection of [GeneralSuggestionModel](#).
Filtered for (suggestion_type= SUGGESTION_TYPE_ADD_QUESTION AND status!=STATUS_IN_REVIEW.
Sort in decreasing created_date with limit Of 100)

question_reviewer_stats_models:- Pcollection of [QuestionReviewStatsModel](#) (Grouped by id(contributor_id))

Transformation:-

For performance metric data

recent_review_outcomes:- append the status

For other fields

topic_ids_with_translation_submissions:- append the topic_id

counts:- sum all the other count entities

Feature Gating

Feature Flag 1: (CD_NEW_UI)

New dashboard will be gated behind this feature flag.

After feature flag is in place, we will implement in *core/templates/pages/contributor-dashboard-admin-page/contributor-dashboard-admin-page.mapage.html* with an if condition. For <contributor-dashboard-admin-page> and <new-contributor-dashboard-admin-page>. The URL for the contributor-dashboard-admin remains the same.

Include TS files of both the components in *core/templates/pages/contributor-dashboard-admin-page/contributor-dashboard-admin-page.import.ts*. TS files for both the components will be included in the Webpack bundle for /contributor-dashboard-admin page.

This flag will be implemented so that the existing contributor-admin-dashboard will still be accessible. Newly implemented UI in milestone 1 will be gated behind this flag. The UI gated behind this flag will be

Old component is <contributor-dashboard-admin-page>.

New component <contributor-admin-dashboard-page> will be created.

Milestone Table (include both PRs and other actions that need to be taken prior to launch)

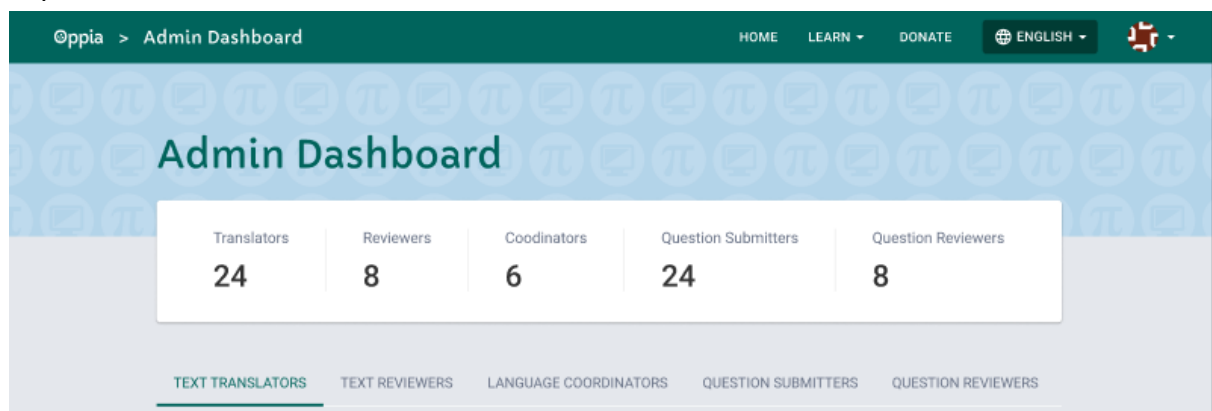
Milestone 1

Key objective for this milestone:

- MileStone 1:- **A Basic UI structure with different tabs will be visible**. Backend storage and service changes will be done. Backend changes will include required changes for pagination and sorting.
- The user can click on different tabs i.e.
 - Translation Submitters(only Language Coordinator)
 - User

- Accuracy
- Performance
- Translated cards
- Last Translated
- Translation Reviewers(only Language Coordinator)
 - User
 - Reviewed Cards
 - Last reviewed
- Question Submitters(only question coordinator)
 - User
 - Accuracy
 - Performance
 - Questions Submitted
 - Last submitted
- Question Reviewers(Only question coordinator)
 - User
 - Question Reviewed
 - Last reviewed

-
- Super Admin will access all the views.



- The user can see the above interface on navigating to /contributor-dashboard-admin.
- The UI delivered will be static and the user can't edit roles.
- Users can only select top level tabs and can't interact with the tables beyond that.
- The tables below tabs will be populated with a chunk of data i.e. 20 entries of each stats table will be displayed.
- The stats displayed won't be in any sorted order rather a subset of the whole data
- Language/Question coordinator tab won't be available yet since we'll be introducing the role of language coordinator in milestone 2.

Note: All flows will work correctly on desktop and mobile.

No.	Description of PR / action	Prereq PR numbers	Target date for PR creation	Target date for PR to be merged
1	Introducing the role of Question Coordinator with rights same as question admin.	-	04/06/2023	08/06/2023
2	Making Storage Model changes to store performance related data (the fields will be empty at this stage).	-	08/06/2023	12/06/2023
3	Introducing all the Domain Objects required for the project. TranslationSubmitterContributionStats TranslationReviewerContributionStats QuestionSubmitterContributionStats QuestionReviewerContributionStats	-	12/06/2023	16/06/2023
4	Writing beam job to populate the existing data for performance measure. (This will need to be tested on a prod server since it involves a data update.)	1.2	21/06/2023	28/06/2023
5	Modifying the existing services as mentioned here and adding services to handle contributor stats.	-	27/06/2023	01/07/2023
6	Implementing handler changes. Controller layer changes to connect services(M1.4) with frontend. Two new handlers LanguageCoordinatorRightsHandler and AllContributorStatsSummariesHandler	1.3, 1. 5	01/07/2023	05/07/2023
7	Creating a basic UI with a static view of all the tabs according to their assigned roles. Feature Flag 1(CD_NEW_UI) Populate a page of stats in tabular format on frontend, for the 4 tabs corresponding to translation/question reviewers and submitters.	1.3, 1.4, 1.5, 1.6 M1.4 needs to be deployed on the prod server.	11/07/2023	14/07/2023
	Meeting with the PM.	1.8	14/07/2023	

Milestone 2

Key objective for this milestone:

- MileStone 2:- A fully functional Contributor Dashboard Admin Stats table with 6 tabs
 - a) Translation Submitters
 - b) Translation Reviewers
 - c) Language Coordinators
 - d) Question Submitters
 - e) Question Reviewers
 - f) Question Coordinators(newly added as discussed with Sean)

Each table will have pagination, filtering and sorting features.

Note: All flows will work correctly on desktop and mobile.

No.	Description of PR / action	Prereq PR numbers	Target date for PR creation	Target date for PR to be merged
1	Make Storage model and handler changes to store and update the language coordinator rights.	-	02/08/2023	06/08/2023
2	Introduce Language Coordinator role with permissions to view and edit Language translators and reviewers. Writing e2e tests for question coordinator and language coordinators to access the /contributor-admin-dashboard and view the stats.	1.6, 2.1	10/08/2023	14/08/2023
3	Making changes to /admin page to promote a user to language/question coordinator or to deassign a user from coordinator rights. Writing e2e tests for web super admin to update coordinator roles in /admin	2.2	16/08/2023	20/07/2023
4	Making different admin actions available according to the roles in /contributor-dashboard-admin (for eg. Promoting a translator to reviewer as a language coordinator)	2.2	23/08/2023	27/08/2023

	Covering e2e tests for admin actions available in /contributor-dashboard-admin			
5	Implementing Pagination for 4 tabs in /contributor-dashboard-admin E2e tests for testing pagination	1.7, 2.4	28/08/2023	02/09/2023
	Meeting with PM	2.5	30/08/2023	
6	Display Language Coordinator and Question Coordinator stats in /contributor-dashboard-admin along with pagination. Writing e2e tests for web super admins to access /contributor-dashboard-admin and view the stats.	1.7, 2.3	03/09/2023	06/09/2023
7	Implementing sorting and filtering for all 6 tabs in /contributor-dashboard-admin. E2e tests for sorting and filtering.	2.6	08/09/2023	12/09/2023
	Meeting with PM	2.7	15/09/2023	
8	Remove the roles of Question Admin and Translation Admin from the system (once existing question admins and translation admins are promoted as question/language coordinators)	1.1 and 2.4 deployed on the server and steps mentioned here are done.	12/09/2023	15/09/2023

Questions to ask to stakeholders during demo:-

- 1) Whether the tooltip on column heading must show weights or the information about the filtering of stats.

Future Work

- **Find a method to analyze the stats of the contributors and make some high level strategic decisions.**
By analyzing the stats of contributors, the coordinators should be able to notify the contributors or take updates from them about the contributors availability. And if the contributor has discontinued or will be busy with some other work. The coordinator can demote them from reviewer again.