Google Summer of Code - 2022

Fix Validation Errors for Explorations

Proposal by - Hitesh Tomar

Section 1: About You

What project are you applying for?

Fix validation errors

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

I'm not sure if I can find a place where I can learn and help at the same time. You are learning and working with some great minds who have the same mindset of helping others. I think Oppia is the most welcoming community I've been part of than any other community out there.
*The reason I choose the **Fix validation errors** project is that I'm solely interested in the backend and got introduced to Apache beam jobs just a few months back and I'm really enjoying learning it.*

Prior experience

I've been working on Apache beam jobs for a few months now, currently I'm leading one of the projects you can take a look at here. Currently, I've been reviewing all of the PRs related to the project.
I've also worked on the backend validations as we will require some for this particular project.
Also, I'm currently doing my internship as a backend developer which is giving me more clarity on how things work in the backend.

Some of my PRs are -

- *Exploration title length should have a max length of 36.*
    - *I did some research work regarding this PR as we wanted to make sure after we add a backend validation regarding this it doesn't break anything. You can find a detailed analysis here.*
- *Validate skill medium rubrics explanations.*
- *List all filepath-with-value state fields as an empty string.*
- *Story nodes and skill rubrics backend validation.*
- *Validation for Continue, Text Input, Multiple choice interaction.*

Project size

large (~350 hours)

Project timeframe

June 13 - November 13

Contact info and timezone(s)

- ***Mob** - +91 8696001333*
- ***Mail** - lkbhitesh07@gmail.com*
- ***Timezone** - Indian Standard Time (GMT +5:30)*
- ***Preferred mode of communication** - Google hangouts, Gmail, WhatsApp (or any mode will also work)*

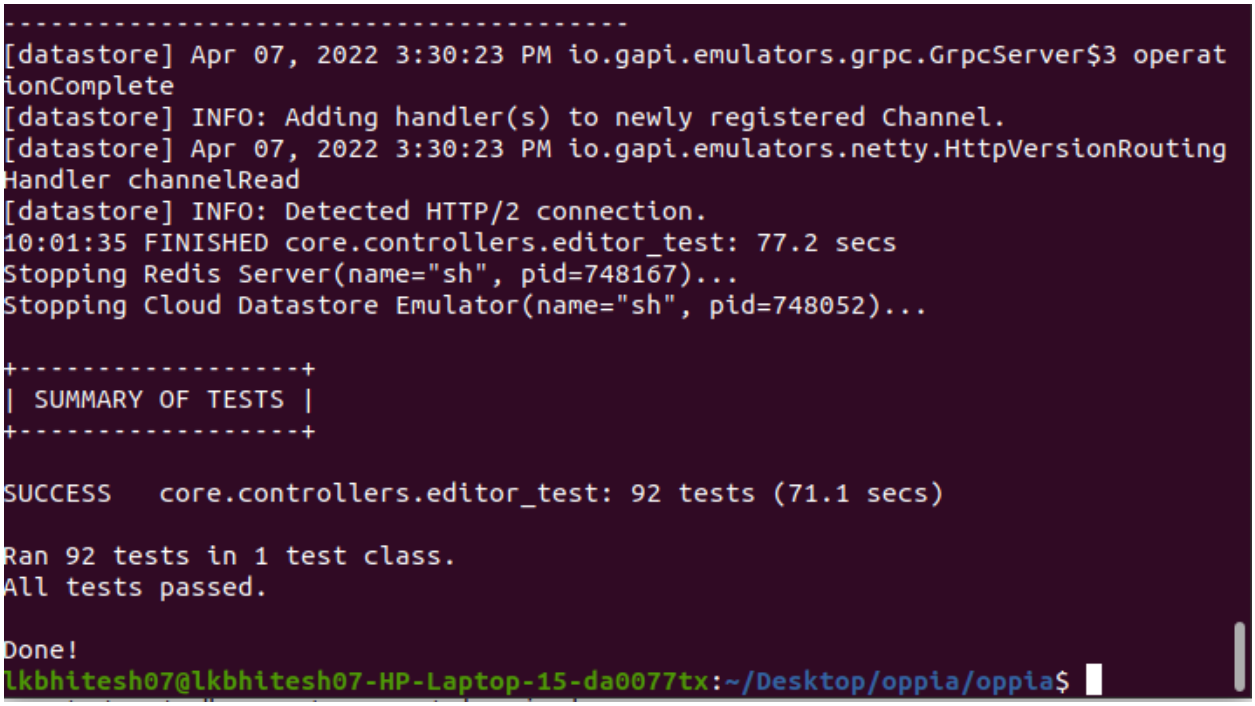
Time commitment

As I will be contributing to the large project, I will be working 20 hrs/week on this project and if required I can totally extend accordingly.

Essential Prerequisites

Answer the following questions (for Oppia Web GSoC contributors):

- *I am able to run a single backend test target on my machine. (Show a screenshot of a successful test.)*



- *I am able to run all the frontend tests at once on my machine. (Show a screenshot of a successful test.)*

```
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7365 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7366 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7367 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7368 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7369 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7370 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7371 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7372 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7373 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7374 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7375 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7376 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7377 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7378 of 7378 SUCCESS (0
Chrome Headless 99.0.4844.51 (Linux x86_64): Executed 7378 of 7378 SUCCESS (2
mins 36.728 secs / 2 mins 13.279 secs)
TOTAL: 7378 SUCCESS
TOTAL: 7378 SUCCESS
07 04 2022 15:48:27.036:WARN [launcher]: ChromeHeadless was not killed in 2000
 ms, sending SIGKILL.
Done!
lkbhitesh07@lkbhitesh07-HP-Laptop-15-da0077tx:~/Desktop/oppia/oppia$
```

- *I am able to run one suite of e2e tests on my machine. (Show a screenshot of a successful test.)*

```
.
  Blog dashboard functionality
    ✓✓✓ should check user profile is visible on both blog dashboard and editor, create, edit and delete a blog post from blog dashboard
[18:00:04] W/element - more than one element found for locator By(css selector, .toast-success) - the first result will be used
.    ✓✓✓ should create, publish, and delete the published blog post from dashboard.
[18:00:22] 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 delete the blog post
[18:00:34] W/element - more than one element found for locator By(css selector, .toast-success) - the first result will be used
[datastore] Apr 07, 2022 6:00:38 PM io.gapi.emulators.grpc.GrpcServer$3 operationComplete
[datastore] INFO: Adding handler(s) to newly registered Channel.
[datastore] Apr 07, 2022 6:00:38 PM io.gapi.emulators.netty.HttpVersionRoutingHandler channelRead
[datastore] INFO: Detected HTTP/2 connection.
.    ✓✓✓ should create multiple blog posts both published and drafts and check for navigation through list view



4 specs, 0 failures
Finished in 179.651 seconds

Executed 4 of 4 specs SUCCESS in 3 mins.
[18:01:18] I/launcher - 0 instance(s) of WebDriver still running
[18:01:18] I/launcher - chrome #01 passed
Stopping Protractor Server(pid=849435)...
Stopping Webdriver manager(name="sh", pid=849311)...
Stopping GAE Development Server(name="sh", pid=847831)...
[2022-04-07 18:01:19 +0530] [849228] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [849991] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [850022] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [849239] [INFO] Worker exiting (pid: 849239)
[2022-04-07 18:01:19 +0530] [850100] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [850194] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [850066] [INFO] Worker exiting (pid: 850066)
[2022-04-07 18:01:19 +0530] [850395] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [849995] [INFO] Worker exiting (pid: 849995)
[2022-04-07 18:01:19 +0530] [850104] [INFO] Worker exiting (pid: 850104)
[2022-04-07 18:01:19 +0530] [851093] [INFO] Handling signal: term
[2022-04-07 18:01:19 +0530] [851097] [INFO] Worker exiting (pid: 851097)
[2022-04-07 18:01:19 +0530] [850198] [INFO] Worker exiting (pid: 850198)
[2022-04-07 18:01:19 +0530] [850461] [INFO] Worker exiting (pid: 850461)
[2022-04-07 18:01:19 +0530] [849991] [WARNING] Worker with pid 849995 was terminated due to signal 15
[2022-04-07 18:01:19 +0530] [850022] [WARNING] Worker with pid 850066 was terminated due to signal 15
[2022-04-07 18:01:19 +0530] [850100] [WARNING] Worker with pid 850104 was terminated due to signal 15
[2022-04-07 18:01:19 +0530] [849228] [WARNING] Worker with pid 849239 was terminated due to signal 15
[2022-04-07 18:01:19 +0530] [849228] [INFO] Shutting down: Master
[2022-04-07 18:01:19 +0530] [849991] [INFO] Shutting down: Master
[2022-04-07 18:01:19 +0530] [850022] [INFO] Shutting down: Master
```

Other summer obligations

I might start my full-time job around August or September. I will be graduating in a few months and I will be looking for jobs but that won't affect the contribution here on the project.
That's the reason I took the long project and will contribute a decent amount of time per week.

Communication channels

I would like to have twice a week interaction with my mentor so that I can update and ask doubts. Any mode of communication will work.

Section 2: Proposal Details

Problem Statement

Link to PRD (or N/A if there isn't one)	N/A
Target Audience	Release Coordinator, Oppia Developers
Core User Need	There are some validation errors that need to be fixed because as a developer or release coordinator when I access the storage layer data or run the jobs, I get several errors. As a learner, the validation errors will result in harder to access the data and will end up lowering my experience on the website.
What goals do we want the solution to achieve?	This will enable us to validate various attributes of our models and fix any consistency errors that are detected. Previously we did not have any validations while storing the data so there is a strong possibility of having some invalid data in our datastore, our end goal is to take out all the invalid data and fix them after that apply the validations so that it does not happen in the future. This project majorly focuses on the errors related to the Explorations which are General State Validations, General RTE Validations, and General Interaction Validations. Additionally, we will have a way to communicate with GCS and perform actions like Read file, Write file, Get all the contents of the folder, and edit the metadata of file. All the audio files will have 'audio/mpeg' as the MIME type. All the profile images in the UserSettingsModel will be migrated to GCS and it's webP will be generated.

Section 2.1: 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)**	**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.**
1.1	Exploration State	Lesson creators should be aware when there is an error in their lesson state, and should not be able to save or publish the exploration until it is fixed. Also, the user should have a great learning experience.	High	Validations for the following: See specific validation in the below section.	N/A
1.2	Exploration Interactions	Lesson creators should be aware when there is an error in their lesson state, and should not be able to save or publish the exploration until it is fixed. Also, the user should have a great learning experience.	High	Validations for the following See specific validation in the below section.	N/A
1.3	Exploration RTE	Lesson creator should be aware when there is an error in their lesson state, and should not be able to save or publish the exploration until it is fixed. Also, the user should have a great learning experience.	High	Validations for the following See specific validation in the below section.	N/A

2	Move and fix data in GCS	It should be easy for developers to fetch data from the GCS, data like audio, and images.	High	• Introduce GCS IO for Beam jobs (should be placed in core/jobs/io), which will allow Beam jobs to work with files in GCS. • Validate that existing files in GCS have the correct MIME types (#13480), and fix those types if needed. • Migrate profile images from UserSettingsModel to GCS and also generate WebP for profile images (does not include frontend changes)	N/A

Section 2.2: HOW

Step 1: Write Beam Job to get all Exploration validation errors

The beam job to collect the validation errors -

1. PR #15563 - This PR validates the following
    a. General state validations
        i. tagged_skill_misconception_id should be None
        ii. The default outcome should have a valid destination node
        iii. destination_id should be non-empty and match the ID of a state in the exploration
        iv. Outcome labelled_as_correct should not be True if destination ID is “(try again)”
        v. The answer group should have at least one rule spec

vi. refresher_exploration_id should be None for all lessons

b. General RTE validations
    i. Image tags contain filepath, alt, and caption attributes, where caption can be an empty string with at most 160
    characters and alt should have at least 5 characters.
    ii. Math tags contain math_content, raw_latex, and svg_filename attributes, where svg_filename of curated exp has an SVG extension.
    iii. Skillreview tags contain text attributes, text is non-empty
    iv. Video tags contain video_id, start, end, and autoplay attributes, where start is before end
    v. Link tags contain text and url attributes, where text is non-empty

c. General interactions validations
    i. Continue
        1. Text should be non-empty and have a max-length of 20.
        2. Should only have a default outcome (and no answer groups) associated with it.
    ii. End Exploration
        1. Should not have a default outcome or any answer groups.
        2. Should be at most 3 recommended explorations Note: crossover with Exploration
    iii. Numeric Input
        1. For x in [a, b], a must not be greater than b
        2. For x in [a-b, a+b], b must be a positive value
    iv. Fraction Input
        1. All rules should have solutions in the simplest form if the simplest form setting is turned on
        2. All rules should have solutions in proper form if the allow improper fraction setting is turned off
        3. Rule 'exactly equals' should have a solution without integer parts when the allow nonzero integer parts setting is turned off
        4. Fractional denominator should be > 0
    v. Number With Units Input
        1. `equal to` should not come after `equivalent to` if they have the same value
    vi. Multiple Choice Input
        1. Answer choices should be non-empty and unique
        2. No answer choice should appear in more than one answer group
        3. If all MC options have feedback, do not ask for a "Default Feedback"
    vii. Item Selection Input
        1. Min number of selections should be no greater than max num

        2. There should be enough choices to have min num of selections
        3. All items should be unique and non-empty
        4. == should have between min and max number of selections
    viii. Drag and Drop Input
        1. All inputs should be non-empty, unique
        2. There should be at least 2 items
        3. Multiple items can be in the same place iff the setting is turned on
        4. `== +/- 1` should never be an option if the "multiple items in same place" option is turned off
        5. for `a < b`, a should not be the same as `b`

2. PR #15748: This PR validates the following
    a. General RTE validation
        i. Every tag should contain there attributes even if they are empty.
    b. General interaction validation
        i. Numeric Input
            1. Each answer group should not be a subset of any answer group that comes before it.
        ii. Fraction Input
            1. All rules should have solutions that do not match previous rules' solutions
        iii. Item Selection Input
            1. None of the answer groups should be the same
        iv. Drag and Drop Input
            1. `==` should come before idx(a) == b if it satisfies that condition
            2. `==` should come before == +/- 1 if they are off by at most 1 value
        v. TextInput
            1. Text Input height should be between integer between 1 and 10, inclusive
            2. contains should always come after any other rule where the contains string is a substring of the other rule's string
            3. starts with should always come after any other rule where a starts with string is a prefix of the other rule's string
        vi. EndInteraction
            1. All recommended explorations should be valid

3. PR #15714: This includes the check from #15563 which required further investigation and the result is categorized in Private, Public and Curated entities. Checks are as follows
    a. refresher_exploration_id should be None for all lessons
    b. Text should be non-empty and have a max-length of 20 (Continue Interaction)

c. == should have between min and max number of selections (ItemSelection Interaction)
d. Multiple items can be in the same place iff the setting is turned on (DragAndDrop Interaction)
e. == +/- 1 should never be an option if the "multiple items in same place" option is turned off (DragAndDrop Interaction)
f. alt should have at least 5 characters (RTE image)
g. Image should have an SVG extension (RTE image)
h. Start value is before end value (RTE video)

4. PR #15172: RTE image should have valid 'filepath-with-value' attribute

   NOTE: Details to this check is present in the "General State RTE validations" in the image part

Step 2: Fix all validation errors

Validation Checks for Exploration State

General state validations

tagged_skill_misconception_id should be None

^ has no errors Frontend validation Backend validation

The default outcome should have a valid destination node

^ has no errors Frontend validation Backend validation

destination_id should be non-empty and match the ID of a state in the exploration

^ has no errors Frontend validation Backend validation

Outcome labelled_as_correct should not be True if destination ID is (try again)

^ has 8 errors Frontend validation Backend validation

- **Description of Errors:** The `labelled_as_correct` value of an answer group is true even when its destination node is the state itself or in other words, the destination id is (try again). If any answer group is marked as correct then its destination node should be in some other state so that the user can move to the next part or card.
- **Why did the error occur in the first place:** Previously we did not have any frontend validation for this and that might be the reason that the creator accidentally marked the answer group as `labelled_as_correct` and still set the destination node as the state itself.
- **Plans to Fix Errors:** Ways to fix the error -

- One way to fix this error would be to simply uncheck the `labelled_as_correct` value or mark the value as False the reason being is if any answer group has a destination id of (try again) it only means that the answer that user has given is wrong and creator wants the user to try again. Future updates of the models will have no effect on the data as we are only modifying the value. I can think of two ways to fix this particular error -
    - To fix it manually - We can totally fix this particular error by manually visiting the exploration and unchecking the value the reason being as we have only 8 errored values.
    - To fix it via job - We can totally write the job and can change the value of the `labelled_as_correct` value. To be specific it is present inside Interaction -> AnswerGroup -> Outcome -> labelled_as_correct
    - Another way to fix this error would be to use a conversion function, we can write a conversion function that will fix the error even if the user reverts back to the errored lesson.
      I will be receiving the exploration states as an argument in the conversion function and as this check should be present in all the 3 entities which are private, public, and curated I will simply filter out the invalid state and assign `labelled_as_correct` as False.

I'm planning to go ahead with the third approach which is to use the conversion function to resolve the error. The problem with the first approach is that the user can revert back to the errored lesson and that way we will have the error again. Even if we use the 'PutModel' approach to fix the error that will only fix the current data in the datastore and if the user reverts back then the data will become invalid again and we do not want that. If we use the second approach which is the conversion function approach then all the current data will be fixed with the help of the state migration job and even when the creator reverts to an invalid lesson the data will be passed by the conversion function and become valid again.

The answer group should have at least one rule spec
^ has no errors Frontend validation Backend validation

refresher_exploration_id should be None for all lessons
^ has 13 errors Frontend validation Backend validation

- **Description of Errors:** The `refresher_exploration_id` for all lessons should be None for the explorations but here we got several states having some value.
- **Why did the error occur in the first place:** We do not have frontend validation for this so that might be the reason why this error has occurred.
- **Plans to Fix Errors:** I think we can simply change the `refresher_exploration_id` to None as it does not make sense when it's present in the Exploration I think this particular field is only for the Question part. Please note that before changing the value to `None` some edits need to be made to the errored explorations like we may need to create new

cards in the exploration to support the remedial part that would now be taken away. I have submitted the list of errored explorations to Sean and he will be fixing it manually. Please note that all of the errored explorations are curated and they will be fixed manually.

Validation Checks for Exploration Interaction

General rules: These are some general rules that are applicable to the interactions below.

1. If the solution to checks is to simply remove the `rule_spec` from the `answer_group` and after removing them if the `answer_group` becomes empty then the `answer_group` will also be removed.
2. There is a strong possibility that in case we remove the `answer_group` and it may result in the state disconnection. So for all the checks where I plan to remove the `rule_specs` and `answer_groups` if appropriate, I ran an audit job to see how many explorations may result in the state disconnection. You can find the job here.
3. Some of the solutions require removing the `rules`, `answer_group` and `choices` from the state and each of them consists of `content_id`. We need to edit the `WrittenTranslations` and `RecordedVoiceovers` sections as they may have the `content_ids` that we have deleted.
   For this we will need to edit the state dictionary, we have 2 fields inside the state which are `written_translations` and `recorded_voiceovers`. I will keep track of all the content ids that I'm going to delete in a state and in the end, I will iterate over these 2 fields and remove those `content_ids` from here.

Continue

Text should be non-empty and have a max-length of 20.

^ has 869 errors Frontend validation Backend validation

- **Description of Errors:** The length of the text value of continue interaction is more than 20.
- **Why did the error occur in the first place:** The reason for this is that we did not have any frontend validation for this part and creators were allowed to have text value of more than 20 characters in length but now we have added the frontend validation for this part.
- **Plans to Fix Errors:** The possible solution I can think of
  - One solution to fix this error would be to simply replace the value with the default one which would be `Continue` for the explorations present in the English language. If we want we can manually edit the curated explorations(if any) and can change the text value accordingly and for all the other explorations we can simply assign the default values.
  - Another possible solution can be to leave it as it is, Now as soon as the creator will edit the exploration our frontend validation will take care of this error, the creator will not be able to save the exploration until unless they edit the text value of Continue interaction.

- One approach would be to simply trim the text value to length 20 but that way the `text` will not make any sense and it will be confusing for the learners.

Method to fix: The safest approach would be to go ahead with the conversion function. I will write the check in the conversion function and filter out all the invalid continue interactions that have text values more than specific and then I will be setting the value to the default, for `en` is 'Continue'. For each of the language code, I will translate `Continue` word to that specific language and replace the text value of the interaction. The errored language codes are -

- es
- en
- nl
- ru
- sr
- bg
- fr
- ca
- hu
- zh
- it
- fi
- pt
- de
- ar
- cs
- tr

I will receive exp state dictionary as an argument to the state conversion function and from there I will iterate and filter out the invalid states and then make them valid.

Should only have a default outcome (and no answer groups) associated with it.

^ has no errors Frontend validation Backend validation

End Exploration

Should not have a default outcome or any answer groups.

^ has no errors Frontend validation Backend validation

All recommended explorations should be valid

^ has 208 errors Frontend validation Backend validation

- **Description of Errors:** End explorations have recommended explorations and they need to be valid which means they should exist and should be public, In our case, we found 208 states that have invalid recommended explorations.

- **Why did the error occur in the first place:** One reason could be that the explorations that are present in the recommended explorations section are now deleted or made private.
- **Plans to Fix Errors:** We are not planning to fix the data that is currently present because even if we do that then more bad data will come eventually as we would not know when an exploration may become private or gets deleted. We do have a robust frontend present as when an exploration is marked private or does not exist, it does not show up to the learner as a recommended exploration. On the lesson creator side, it raises a validation error. So as a result nothing needs to be done here as this will be handled by the creator itself.

Should be at most 3 recommended explorations

^ has 10 errors Frontend validation Backend validation

- **Description of Errors:** End exploration should only have 3 recommended explorations but got more than that.
- **Why did the error occur in the first place:** Previously we did not have any frontend validation regarding this so creators might have added more than 3 recommended explorations but now we have a robust frontend validation for the same.
- **Plans to Fix Errors:**
    - As these are only the recommended explorations we can simply remove the last few to make the count 3.
    - Another possible solution can be to leave it as it is, Now as soon as the creator will edit the exploration our frontend validation will take care of this error, the creator will not be able to save the exploration until unless they edit the number of recommended explorations.

  Method to fix: I will be using the conversion function to fix the error. The approach would be simply to filter all the end explorations and then count the number of recommended explorations it has, if the value is more than 3 we can simply remove the extra values from the list.

Numeric Input

For x in [a, b], a must not be greater than b

^ has 1 error Frontend validation Backend validation

- **Description of Errors:** The rule type `IsInclusivelyBetween` has values `a` and `b` which tells the user if the answer is between `a` and `b`, we found out that the value of `b` is greater than the value of `a` and that should not be valid.
- **Why did the error occur in the first place:** Previously we did not have any frontend validation regarding this so creators might have mistakenly assigned wrong values, I guess that's why we have only 1 error.
- **Plans to Fix Errors:**

- One approach to resolve this error would be to simply do it manually so that we can take a look at the whole exploration and make the values correct which means simply swapping the value.
- Another approach would be to write the job to resolve the error and we can simply swap the values of `a` and `b` because that would make more sense.
- The safest approach would be to go ahead with the conversion function. I will add this check inside the state conversion function and if any value has `a` greater than `b` I will simply swap those values.

  I'm planning to use the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

For x in [a-b, a+b], b must be a positive value
^ has 1 error Frontend validation Backend validation

- **Description of Errors:** The rule type `IsWithinTolerance` have two values which are `x` and `tol` which simply tells if the answer of the user is between `x+tol` and `x-tol` and it is mandatory that the value of the `tol` should not be negative.
- **Why did the error occur in the first place:** We do not have frontend validation for this part and that's the reason we might have encountered this errored value which is the negative value of `tol`.
- **Plans to Fix Errors:**
  - One approach to resolve this error would be to simply do it manually so that we can take a look at the whole exploration and make the values correct which means simply we can make the value positive the reason being is that the range will not change because we are calculating the range by both subtracting and adding the `tol` value.
  - Another approach would be to write the job to resolve the error and we can simply make the `tol` value positive.
  - The safest approach would be to go ahead with the conversion function. I will add this check inside the state conversion function and if any value has `tol` value as negetive I will simply make it positive. If the value will be 0, I will simply convert the rule to the `Equals` rule. (Though I will check if a similar rule already exists first – if so, then I will just delete the tol=0 rule.)

    I'm planning to use the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Note - We only have one error for this check and that exploration was a test exploration which is now deleted.

`IsLessThanOrEqualTo` rule contains string value
^ has 30 errors Frontend validation Backend validation

- **Description of Errors:** The rule `IsLessThanOrEqualTo` should only contain int or float values but it contains string values in our case.
- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this so the string values might got saved in the datastore.
- **Plans to Fix Errors:** To fix this, the optimal way would be to simply remove the rule because we should not allow string values to the numeric input interaction. Remove the answer groups if no rules are left. We will not be able to convert the string values to float or int because the values that are present is of something like `{{Magician Number}}`
- **Method to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here.

I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Note - All the rules present in the `NumericInput` interaction which has a string value, I tend to delete them. To check if the deletion of invalid rules does not result in the disconnection of the current state to the next state I wrote a job, you can find that here. The errored explorations are as follows, though a strange thing happened that all these explorations contain same state names and the same errored answer group and rule specs -
- 096K9qLLmvQi, {State_names - "PlayerGuessing", "Player Guess Setup", "practice ranges", "worst case bigger", "Worst Case Guess", "worst case smaller", "practice ranges lowest"}

- rMFcJ8LO2npk
- y7RCIl0Rn-Tv
- 2
- 9buUiVlCQxPv
- HQKR8LTzJoII
- znJCURVhZ0j8
- 1F3igZt4mpVT
- VyOM3LNvTS6g

- N9x-GUMCS6fT

The `2` exploration was public and all the other explorations follows the cloned_from property and are clone of the `2` exploration and that is why the errored states are same for all the explorations. The `2` exploration is now unpublished.
If the state disconnection happens to a private exploration then it will probably be okay as the creator will not be able to publish it until unless the creatore resolves it.
Please note that all the explorations are private now so it will be fine if state disconnection happens.

`IsGreaterThanOrEqualTo` rule contains string value
^ has 60 errors Frontend validation Backend validation

- **Description of Errors:** The rule `IsGreaterThanOrEqualTo` should only contain int or float values but it contains string values in our case.
- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this so the string values might got saved in the datastore.
- **Plans to Fix Errors:** To fix this, the optimal way would be to simply remove the rule because we should not allow string values to the numeric input interaction. Remove the answer groups if no rules are left. We will not be able to convert the string values to float or int because the values that are present is of something like `{{Magician Number}}`
- **Method to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here.

I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`Equal` rule contains string value
^ has 70 errors Frontend validation Backend validation

- **Description of Errors:** The rule `Equal` should only contain int or float values but it contains string values in our case.
- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this so the string values might got saved in the datastore.
- **Plans to Fix Errors:** To fix this, the optimal way would be to simply remove the rule because we should not allow string values to the numeric input interaction. Remove the

answer groups if no rules are left. We will not be able to convert the string values to float or int because the values that are present is of something like `{{Magician Number}}`

- **Method to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here.

    I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`IsLessThan` rule contains string value

^ has 90 errors Frontend validation Backend validation

- **Description of Errors:** The rule `IsLessThan` should only contain int or float values but it contains string values in our case.
- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this so the string values might got saved in the datastore.
- **Plans to Fix Errors:** To fix this, the optimal way would be to simply remove the rule because we should not allow string values to the numeric input interaction. Remove the answer groups if no rules are left. We will not be able to convert the string values to float or int because the values that are present is of something like `{{Magician Number}}`
- **Method to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here.

    I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`IsGreaterThan` rule contains string value

^ has 60 errors Frontend validation Backend validation

- **Description of Errors:** The rule `IsGreaterThan` should only contain int or float values but it contains string values in our case.

- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this so the string values might get saved in the datastore.
- **Plans to Fix Errors:** To fix this, the optimal way would be to simply remove the rule because we should not allow string values to the numeric input interaction. Remove the answer groups if no rules are left. We will not be able to convert the string values to float or int because the values that are present is of something like `{{Magician Number}}`
- **Method to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here.

    I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

All rules should have solutions that do not match previous rules' solutions

^ has 150 errors Frontend validation Backend validation

- **Description of Errors:** The answer group should not be a subset of any answer group that comes before it because as it will never be matched. This means if any rule spec comes in the range of another rule spec then it will never going to get matched. For example, if we have a rule `IsLessThanOrEqualTo` having value 10 and we have another rule `Equals` having a value 5, then the `Equals` rule will never going to get matched. Currently all of the rule specs that are present in the interaction can cause this error by intersecting each others range, the rule specs are -
    - `IsLessThanOrEqualTo`
    - `IsGreaterThanOrEqualTo`
    - `IsLessThan`
    - `IsGreaterThan`
    - `Equals`
    - `IsWithinTolerance`
    - `IsInclusiveBetween`

    To be specific rules that can intersect the ranges can be -
    - `IsLessThanOrEqualTo` -> IsLessThan, Equals, IsWithinTolerance, IsInclusiveBetween
    - `IsGreaterThanOrEqualTo` -> IsGreaterThan, Equals, IsWithinTolerance, IsInclusiveBetween
- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this.

- **Plans to Fix Errors:** I will be simply be removing the answer group which is subset of the answer group that comes before it. We can simply remove it because it will never going to match anyways.
- **Methods to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here. I will simply be removing the invalid answergroup or the invalid rule spec inside it.

    I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Fraction Input

All rules should have solutions in the simplest form if the simplest form setting is turned on

^ has no errors Frontend validation Backend validation

All rules should have solutions that do not match previous rules solutions

^ has 96 errors Frontend validation Backend validation

- **Description of Errors:** All the rules present inside the answer groups should have a solution that does not match the previous rules solution, if it does the rule will become redundant and will never be matched. This means if any rule spec comes in the range of another rule spec then it will never going to get matched. We have some rule specs like `IsLessThan`, 'IsGreaterThan', 'Equals', so for eg if we have one check - `has fractional part less than 5/2`, then `has fractional part equals 3/2` should come before it otherwise it will not going to match.
  The different cases where this error can occur will be -
    - `HasDenominatorEquals` rule comes before the following -
        - `HasFractionalPartExactlyEquals`
    - `IsLessThan` having some range, and these rule specs are present within that range then the rules will never be matched -
        - `IsEquivalentToAndInSimplestForm`
        - `IsExactlyEqualTo`
        - `IsGreaterThan`
        - `IsEquivalentTo`
    - `IsGreaterThan` having some range, and these rule specs are present within that range then the rules will never be matched -
        - `IsEquivalentToAndInSimplestForm`
        - `IsExactlyEqualTo`

        - `IsLessThan`
        - `IsEquivalentTo`
- **Why did the error occur in the first place:** The reason might be the frontend validation, previously we did not have frontend validation for this.
- **Plans to Fix Errors:** I will be simply be removing the rules which cannot be matched.
- **Methods to fix:**
    - One approach would be to write the job to resolve the error and we can simply remove the rule.
    - Another approach would be to use the conversion function and that would be more optimal here. I will simply be removing the invalid answer group or the invalid rule spec inside it.

    I'm planning to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

All rules should have solutions in proper form if the allow improper fraction setting is turned off
^ has no errors Frontend validation Backend validation

Rule 'exactly equals' should have solution without integer parts when the allow nonzero integer parts setting is turned off
^ has no errors Frontend validation Backend validation

Fractional denominator should be > 0
^ has no errors Frontend validation Backend validation

Number With Units Input

`equal to` should not come after `equivalent to` if they have the same value
^ has no errors Frontend validation Backend validation

Multiple Choice Input

Answer choices should be non-empty and unique
^ has 7 errors Frontend validation Backend validation

- **Description of Errors:** We know that in multiple-choice input we have several options or choices from which we select and it would not make any sense it any of the choices are empty or duplicate and we found some choices having the errored values. We have 6 error places where the choices are duplicates and we have 1 error place where the choice is empty.

- **Why did the error occur in the first place:** Previously we did not have any frontend validation regarding this so creators might have mistakenly added the empty and duplicate values but we now have a robust frontend validation for this purpose.
- **Methods to fix errors:**
    - One approach to resolve this error would be to simply do it manually so that we can take a look at the whole exploration and make the values correct which means simply we can remove the choices which are empty and duplicate.
    - Another approach would be to go ahead and write the job to resolve this error in which we can simply remove the empty and duplicate values.
    - The safest approach would be to use the conversion function to perform this. For duplicate values there are two possibilities - one is that the values are not empty and we can simply remove the latter choice. Another possibility is that the values are empty, in that case, I will simply be replacing the choices with "Choice 1" / "Choice 2". If only one choice is present and is empty I will simply be removing that. Please note that I will also remove the rules which were associated with the removed choices, for that I ran an audit to check if that can result in the state disconnection, I found that there is one exploration in which this can happen and the details are, exp_id - X8cVxz-mQo9_ -> {'state_name': 'Instrument Design', 'ans_group_idx': [0]}. There are 3 empty choices present inside this, I will simply be resolving it with the "Choice 1" / "Choice 2" method.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

No answer choice should appear in more than one answer group

^ has 81 errors Frontend validation Backend validation

- **Description of Errors:** We know that no answer choice can appear in multiple answer group as that particular choice will have multiple destination nodes and which is not practically possible and we have found some choices which are present in more than one answer group.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation regarding this so creators might have mistakenly added the multiple answer group for a single choice but now we have frontend validation for this part.
- **Plans to Fix Errors:** If the choice is present in more than one answer group then all the other answer groups will be redundant and will never be matched and I think it will be safe to simply remove that rule spec from the answer group and in case that answer group has only one rule spec we can simply remove the complete answer group, to be specific we will be removing the later rule_spec which got repeated.
- **Methods to solve:**
    - I can use beam job to resolve the error.

- I'm planning to go ahead with the conversion function approach. With the help of the conversion function I can simply filter out the invalid values which is the duplicate rule_specs and I will simply be removing the later rule_spec and the answer group if no rules are left. The reason to use the conversion function is that even when the creator reverts to the invalid lesson the data will still be passed from the conversion function and it will become valid again.

If all MC options have feedback, do not ask for a "Default Feedback"
^ has 6069 errors Frontend validation Backend validation

- **Description of Errors:** If all the choices are provided with the feedback then there is no need of the "Default Feedback" and we have found several places where we have feedbacks for every choice and still we have default feedback.
- **Why did the error occur in the first place:** The reason behind this is that we do not have frontend validation as of now also the "Default Feedback" section is predefined as soon as you select the interaction, you only have to add "What you'll say to the learner in case of default feedback". I think as soon as we'll implement the frontend validation this issue will be resolved.
- **Plans to Fix Errors:**
    - One approach can be to simply remove the default outcome from the interaction.
    - Another approach can be to leave the default outcomes as it is and robust our frontend, which will be our safest bet. If the creator edits the exploration in future they will be able to see the frontend validation error and will remove the default feedback.

Item Selection Input

Min number of selections should be no greater than max num
^ has 1 error Frontend validation Backend validation

- **Description of Errors:** The minimum number of item selections at one time cannot be greater than the maximum number of item selections at one time. We found one case where this happens.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation and that could be the reason that the creator might have mistakenly put up the wrong values.
- **Plans to Fix Errors:** The good part is that we only have 1 error and that can be resolved easily. Now the best approach would be to simply replace the `min value` and `max value`.
- **Methods to Fix:**
    - One approach would be to manually perform this action by simply visiting the exploration and swapping both the values, that way we will be able to look at the answer groups and rule specs and can check if we are performing it in the right way or not.

- Another approach would be to simply write the job to perform this task and in the job will be simply swapping the value.
- The safest approach would be to go ahead with the conversion function and fix the error. I will simply be going through each states and checking on the ItemSelectionInput interaction for the `max` and `min` values and if the `min` is greater than the `max` I will simply swap those values.

  I would be using the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

There should be enough choices to have min num of selections

^ has 2 errors Frontend validation Backend validation

- **Description of Errors:** The total choices we have are less than the total of minimum number of item selections at one time(`min value`). It do not make sense as we do not have enough choices to select for `min value`.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation and that could be the reason that the creator might have mistakenly put up the wrong values.
- **Plans to Fix Errors:** The good part is that we only have 2 errors so that it will be easy for us to resolve them.
  If we will take a look at the errored values we'll get to know that the number of choices is only 1 and the `min value` is 2, so I think we can simply change the `min value` to 1 though it is strange that we only have one choice in Item Selection input.
- **Methods to fix:**
  - One approach would be to manually perform this action by simply visiting the exploration and change the `min value` which according to me is more efficient way to resolve this error.
  - Another approach would be to simply write the job to perform this task and in the job will simply change the value to 1.
  - Another approach to fix the error would be to go ahead with the conversion function and fix the error. I will simply be going through each state and checking on the ItemSelectionInput interaction for the check. If I find the number of choices less than the number of min value, I will simply set the min value to 1.

    I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

All items should be unique and non-empty

^ has 1 error Frontend validation Backend validation

- **Description of Errors:** The choices we have should not be empty or duplicate that will be invalid, in our case we got one error in which we do have a duplicate value.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation and that could be the reason that the creator might have mistakenly put up the duplicate values.
- **Plans to Fix Errors:** The good part is that we only have 1 error so it will be easy for us to resolve.
  The best approach to resolve this error would be to simply remove the duplicate value as simply that value makes no sense.
  - One approach would be to manually perform this action by simply visiting the exploration and changing the choices by removing the duplicate value which according to me is a more efficient way to resolve this error.
  - Another approach would be to simply write the job to perform this task and in the job will simply be removing the duplicate value, which to be specific will be present in the State -> Interaction -> customization_args -> choices -> values
  - Another approach to fix the error would be to go ahead with the conversion function and fix the error. For duplicate values there are two possibilities - one is that the values are not empty and we can simply remove the latter choice. Another possibility is that the values are empty, in that case I will simply be replacing the choices with "Choice 1" / "Choice 2". If only one choice is present and is empty I will simply be removing that. Please note that I will also remove the rules which were associated with the removed choices.

    I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

  NOTE - Please note that it only occurs in one exploration and after looking at it we found that it does not have any answer group only default outcome was present so there is no chance of state disconnection. In general if this occurs then I will not be deleting the rules, will check for the state disconnection and do the manual work where required.

== should have between min and max number of selections

^ has 146 errors Frontend validation Backend validation

- **Description of Errors:** Rule spec of type `Equals` can have multiple choices present and now those choices should be more than `min value` which means the minimum number of selection at one particular time and should be less than `max value` which is maximum number of selection to one particular time. We found several errors violating the condition.

- **Why did the error occur in the first place:** Previously we did not had any frontend validation and that could be the reason that the creator might have mistakenly put up the more or less values.
- **Plans to Fix Errors:**
    - To resolve this error one possible solution would be to simply change the `min value` or `max value`, we cannot make changes in the choices part because that might result in an invalid answer group or the whole state. So if the number of choices present is less than the `min value` then we can simply change the `min value` to the number of choices present or if it is greater than the `max value` then we can change the `max value` to number of choices present there. This solution may go against the creators intend.
    - Another possible solution would be to simply go ahead and delete the rule and the answer group if only one rule was present, I think this would be the safest approach to go ahead with.
    - One solution would be to simply wait and let the frontend handle this part because as soon as the creator will open the exploration for edit purposes then the frontend will show the errors and it will be mandatory for the creator to resolve that inorder to save the exploration. This will not be the correct way to resolve this as we don't if the exploration is going to get edited and if so then when its going to get edited.
- **Methods to fix:**
    - One solution would be to use the beam job to perform the opertation in which I will simply be deleting the invalid rule.
    - To use the conversion function to perform the task, we can simply filter out the invalid rule and can delete it.

  I'm planning to go ahead and use the second plan to fix the error which is to simply delete the rule. I will be using the conversion function to perform the task as this would be the more optimal one, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

None of the rules should be the same

^ has 70 errors Frontend validation Backend validation

- **Description of Errors:** None of the rules should be same as it will lead to redundancy, in our case we have `Equals` rule which have same values at multiple places.
- **Why did the error occur in the first place:** Previously we did not had frontend validation so that might be one reason.
- **Plans to Fix Errors:** To fix the error I'm planning to simply go ahead and remove the later rule as it will never going to match.
- **Methods to fix:**

- One solution would be to use the beam job to perform the operation in which I will simply be deleting the invalid rule.
- To use the conversion function to perform the task, we can simply filter out the invalid rule and can delete it.

I'm planning to go ahead and use the second method to fix the error which is to use the conversion function and simply remove the errored rule, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Drag and Drop Input

All inputs should be non-empty, unique

^ has no errors Frontend validation Backend validation

There should be at least 2 items

^ has no errors Frontend validation Backend validation

Multiple items can be in the same place iff the setting is turned on

^ has 13 errors Frontend validation Backend validation

- **Description of Errors:** Multiple items or choices should not be in the same place when `Allow multiple items at the same place` setting is turned off and we have found 13 errors that are violating this condition.
- **Why did the error occur in the first place:** Previously we did not have any frontend validation and that could be the reason that the creator might have mistakenly forgotten to click the checkbox.
- **Plans to Fix Errors:** There are a total of 4 explorations containing 13 errors and that is a good part as we will be able to resolve them quickly.
    - One possible solution would be to simply click on the checkbox and mark it as True, here the checkbox refers to `Allow multiple items at the same place`. This way we can simply make the error resolves and we will not require any changes in the answer group or rule specs. This solution may entail changing the intent of the question.
    - Another solution would be to simply delete the rules which do not follow this check. I would like to go ahead with this solution.

- **Methods to fix:**
    - One approach would be to simply visit these 3 explorations and remove the errored rule spec manually.
    - Another way would be to write the job and remove the errored rule spec.
    - The most efficient approach would be to simply go ahead and use the conversion function to delete the invalid rule.

I'm planning to go ahead and use the conversion function method to resolve the error by simply removing the rule spec, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`== +/- 1` should never be an option if the "multiple items in same place" option is turned off

^ has 53 errors Frontend validation Backend validation

- **Description of Errors:** Rule spec of type `IsEqualToIrderingWithOneItemAtIncorrectPosition` should not be present when `Allow multiple items at same place` setting is turned off, we have found some places which violates this condition.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation and that could be the reason that the creator might have mistakenly forgotten to click the checkbox.
- **Plans to Fix Errors:**
    - One way to resolve this error is to simply click on the checkbox or make the value True because we cannot simply change the values of rule specs or answer groups that might result in an invalid answer group or maybe the whole state. This approach may change the intent of the creator.
    - One another approach would be to simply remove the invalid rule. I'm planning to go ahead with this approach.
- **Methods to fix:**
    - I can use beam job to simply remove the invalid rule.
    - I can use conversion function to complete the operation which is to simply go ahead and remove the invalid rule.

  I'm planning to use the conversion function method to resolve this issue, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

for `a < b`, a should not be the same as b

^ has 2 errors Frontend validation Backend validation

- **Description of Errors:** Rule spec of type `HasElementXBeforeElementY` should not have the value `X` equals to the value `Y` because that would not make any sense and it wouldn't be possible to perform this.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation and that could be the reason that the creator might have mistakenly added the same value at `X` and `Y`.
- **Plans to Fix Errors:** The good part is that we only have 2 errors so it will be easy to resolve them.

To fix this particular error I think we can simply remove the rule spec from the answer group as the rule spec makes no sense and will never be matched.

- **Methods to fix:**
    - One approach would be to resolve this error manually, simply we can visit the exploration and move to the desired state and can remove the rule spec from there.
    - Another approach would be to write the job to resolve the error, we can simply visit the answer group and the rule spec present there and can simply remove the rule spec.
    - Another approach to fix the error would be to go ahead with the conversion function and fix the error. I will simply be going through each states and checking on the DragAndDropInput interaction for the check. When I'll find the errored rule_spec I will simply be deleting it.

      I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

== should come before idx(a) == b if it satisfies that condition

^ has 2 errors Frontend validation Backend validation

- **Description of Errors:** `Equals` rule should always come before the `HasElementXatPositionY` rule otherwise it will become redundant and will never be matched.
- **Why did the error occur in the first place:** Previously we did not had frontend validation for this.
- **Plans to Fix Errors:** I'm planning to simply remove the `Equals` rule as it is never going to match.
- **Methods to fix:**
    - We can manually visit the exploration and remove the `Equals` rule from there.
    - I can write beam job to remove the `Equals` rule.
    - I can use conversion function to resolve the error, I will simply remove the equals rule.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

== should come before == +/- 1 if they are off by at most 1 value

^ has 6 errors Frontend validation Backend validation

- **Description of Errors:** `Equals` rule should always come before the `IsEqualToIrderingWithOneItemAtIncorrectPosition` rule otherwise it will become redundant and will never be matched.
- **Why did the error occur in the first place:** Previously we did not had frontend validation for this.
- **Plans to Fix Errors:** I'm planning to simply remove the `Equals` rule as it is never going to match.
- **Methods to fix:**
    - We can manually visit the exploration and remove the `Equals` rule from there.
    - I can write beam job to remove the `Equals` rule.
    - I can use conversion function to resolve the error, I will simply remove the equals rule.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`IsEqualToOrdering` rule have empty values

^ has 39 errors Frontend validation Backend validation

- **Description of Errors:** `IsEqualToOrdering` rule have empty values which it shouldn't.
- **Why did the error occur in the first place:** Previously we did not had frontend validation for this.
- **Plans to Fix Errors:** I'm planning to simply remove the `IsEqualToOrdering` rule as it is never going to match.
- **Methods to fix:**
    - We can manually visit the exploration and remove the `IsEqualToOrdering` rule from there.
    - I can write beam job to remove `IsEqualToOrdering` rule.
    - I can use conversion function to resolve the error, I will simply remove the `IsEqualToOrdering` rule.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Text Input

Text Input height should be integer between 1 and 10, inclusive

^ has 64 errors Frontend validation Backend validation

- **Description of Errors:** Text row height should be between 1 and 10 inclusive.

- **Why did the error occur in the first place:** Previously we did not had frontend validation for this.
- **Plans to fix:** To reduce the text row height to 10.
- **Methods to fix:**
    - I can use beam job to fix the issue, will simply change the row value to 10.
    - I can use the conversion function to resolve the error.

    I'm planning to go ahead with the conversion function approach, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`contains` should always come after the `Equals_without_taking_case_into_account`, `Starts-with` and `Contains` rule where the contains rule string is a substring of the other rule's string

^ has 125 errors Frontend validation Backend validation

- **Description of Errors:** `contains` rule should always come after the `Equals`, `Contains` and `Starts-with` rule otherwise it will become redundant and will never be matched.
- **Why did the error occur in the first place:** Previously we did not had frontend validation for this.
- **Plans to Fix Errors:** I'm planning to simply remove the rule that have `contains` rule substring rule as it is never going to match.
- **Methods to fix:**
    - I can write beam job to simply go ahead and remove the contains rule spec.
    - I can use conversion function to resolve the error, I will simply remove the errored rule.

    I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

`starts-with` should always come after `Equals_without_taking_case_into_account` and `starts_with` rule where a `starts-with` string is a prefix of the other rule's string

^ has 47 errors Frontend validation Backend validation

- **Description of Errors:** `starts with` rule should always come after the `Equals` and `starts-with` rule where a starts-with string is prefix of other mentioned rule's string otherwise it will become redundant and will never be matched.
- **Why did the error occur in the first place:** Previously we did not had frontend validation for this.

- **Plans to Fix Errors:** I'm planning to simply remove the rule that have prefix of the `starts with` rule as it is never going to match.
- **Methods to fix:**
    - I can write beam job to simply go ahead and remove the contains rule spec.
    - I can use conversion function to resolve the error, I will simply remove the errored rule.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Validation Checks for Exploration RTE

RTE tags

Image tags

Contain filepath, alt, and caption attributes, where caption can be an empty string with at most 500 characters and alt should have at least 5 characters also image should be an svg extension.

caption can be an empty string with at most 500 characters

^ has no errors Frontend validation Backend validation

alt should have at least 5 characters

^ has 2842 errors Frontend validation Backend validation

- **Description of Errors:** Image tag having attribute `alt-with-value` should have more than 5 characters and at some places, we found that we are violating this condition. It's simply the alternate value in case we are not able to display the image.
- **Why did the error occur in the first place:** We do not have frontend validation for this part and that's why the creator may have leave the value empty or maybe less than 5 characters.
- **Plans to Fix Errors:**
    - One way to fix the error would be to simply replace the alt value with some default value and I think `Image` would be the right word here for English explorations. This will make a mockery of the alt tag and will not help users who have visual disabilities.
    - Another way to fix would be fix the alt tag manually for the curated explorations and let the backend validation fail for the private and public ones so that when the creator will edit the exploration they will be able to resolve that. I'm planning to go ahead with this fix. I have submitted the list of curated explorations to Sean and he will be fixing them manually.
- **Methods to fix:**

- Will be done manually for the curated explorations and will add the backend validation to prevent it from happening in the future.

Image should have an SVG extension for curated exps
^ has 52 errors Frontend validation Backend validation

- **Description of Errors:** Image tag having attribute `filepath-with-value` should have an svg extenstion but we have found some places where this condition gets violated. `filepath-with-value` simply contains the image.
- **Why did the error occur in the first place:** We do not have frontend validation for this part and that's why the creator may have mistakenly uploaded the image with some other extensions. But today if any creator adds an image then it will be mandatory to upload the file with SVG extension. To be clear we do not have any frontend validation which can detect the extension of the image incase any previously created exploration is edited.
- **Plans to Fix Errors:** It is mandatory that all the curated lessons have the image in SVG format and all the other explorations can have any other format. So currently an art workstream is going on regarding this purpose and I can simply collect the details regarding the image like Exploration, State, etc and can submit it there. I will be coordinating with Sean and Namrata on this, as it needs to be fixed manually.
- After the images are fixed I will add the backend validation so that no new errored data will be formed.

alt-with-value should be an attribute present inside the image tag.
^ has 1041 errors Frontend validation Backend validation

- **Description of Errors:** `alt-with-value` attribute is not present inside the image tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** For the curated explorations no explorations were reported that do not have `alt-with-value` tag inside the image tag and for the public and private I was planning to add the attribute inside the tag and same as the above check I will let the backend fail for this so that the creator can fix it when editing the exploration.

filepath-with-value should be an attribute present inside the image tag.
^ has 32 errors Frontend validation Backend validation

- **Description of Errors:** `filepath-with-value` attribute is not present inside the image tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** Simply planning on delete the tag as we have no information of the image. I will be using the conversion function to perform this task so that even if user reverst back to invalid state, the invalid data becomes valid.

Math tags

Contain math_content, raw_latex, and svg_filename attributes, where svg_filename has an SVG extension.

raw_latex value should not be empty or None
^ has 13 errors Frontend validation Backend validation

- **Description of Errors**: Math tag containing `raw_latex` attribute should not be empty or None but we have found few places where the value is empty.
- **Why did the error occur in the first place:** Previosuly we did not had any frontend validation for the same but it is strange that without the `raw_latex` value we have the SVG image because what we write on `raw_latex` is then gets converted to the svg.
- **Plan to Fix Errors**: To resolve the error I will simply be deleting the bad tags as the tag wouldn't make any sense.
- **Methods to fix:**
    - One approach would be to simply perform the action manually.
    - Another approach would be to use the beam job to resolve the error.
    - The safest approach would be to use the conversion function and in the conversion function, after filtering out the invalid math RTE tags which have empty raw_latex value I will simply delete the tag.

    I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

svg_filename has an SVG extension
^ has no errors Frontend validation Backend validation

Skillreview tags

Skillreview tags contain text attributes, text is non-empty
^ has 1 error Frontend validation Backend validation

- **Description of Errors**: Skillreview tag containing text attribute should not be empty but we have found one place where the attribute value is empty.
- **Why did the error occur in the first place:** We do not have any frontend validation for the same which might be the reason why this error has occurred.
- **Plan to Fix Errors**: The best part is that we only have 1 error so it will be easy for us to fix it.
  The best way to fix this can be to simply remove the tag.
- **Methods to fix:**
    - One approach would be to simply perform the action manually.
    - Another approach would be to use the beam job to resolve the error.
    - The safest approach would be to use the conversion function and in the conversion function, after filtering out the invalid skillreview RTE tags which have

empty text value I will simply delete the tag.

I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Video tags

Video tags contain video_id, start, end, and autoplay attributes, where start is before end

start-with-value should be an attribute inside video tag

^ has 49 errors Frontend validation Backend validation

- **Description of Errors:** `start-with-value` attribute is not present inside the video tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** I'm simply planning to introduce the attribute inside the tag and assign the default value to it which is `0`.
- **Methods to fix:**
    - One approach would be to use the beam job to resolve the error.
    - I'm planning to use the conversion function to resolve the issue.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

end-with-value should be an attribute inside video tag

^ has 49 errors Frontend validation Backend validation

- **Description of Errors:** `end-with-value` attribute is not present inside the video tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** I'm simply planning to introduce the attribute inside the tag and assign the default value to it which is `0`.
- **Methods to fix:**
    - One approach would be to use the beam job to resolve the error.
    - I'm planning to use the conversion function to resolve the issue.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

autoplay-with-value should be an attribute inside video tag

^ has 136 errors Frontend validation Backend validation

- **Description of Errors:** `autoplay-with-value` attribute is not present inside the video tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** I'm simply planning to introduce the attribute inside the tag and assign the default value to it which is `False`.
- **Methods to fix:**
    - One approach would be to use the beam job to resolve the error.
    - I'm planning to use the conversion function to resolve the issue.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Start value is before end value

^ has no errors Frontend validation Backend validation

video_id should be valid

^ has 19 errors Frontend validation Backend validation

- **Description of Errors**: Video tag contains `video_id` attribute which has the video id which creator wants the learner to see. `video_id` should be valid and non-empty but at some places, we found out that `video_id` is empty.
- **Why did the error occur in the first place:** Previously we did not had any frontend validation regarding the same So that might be the reason why this has happened.
- **Plan to Fix Errors**: The main purpose of the video tag is to present the video and that happens via the `video_id` and if that particular attribute is empty then it does not make any sense to keep this tag. I think one solution can be to simply remove the video tag and that way we will be able to resolve this error.
- **Methods to fix:**
    - I will write a beam job in which after filtering out all the invalid video tags and removing them.
    - The safest approach would be to use the conversion function and in the conversion function, after filtering out the invalid video RTE tags which have empty video_id value I will simply delete the tag.

      I would like to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Autoplay attribute should be boolean

^ has 627 errors Frontend validation Backend validation

- **Description of Errors**: The video tag has an attribute named `autoplay` which represents if we want the video in the autoplay mode and this is a boolean field. We have found some places where the value is not boolean.
- **Why did the error occur in the first place:** I'm not sure about this part because currently, we have a checkbox which asks if we want our video to autoplay or not.
- **Plan to Fix Errors**: I think the safe play would be to simply assign the default value to all of the invalid attributes which is `false`, this way we will be able to resolve the error. Before assigning the default value I will be checking if the value in any way meant to be `True` or `False` and will be assigning that value only. So the method to check this would be as after fetching the data we will get that in the string format. I will strip that string value and check if that somehow matches "true" or "false", other than that if incase I recieve values like "True" or "False", I will try to convert those value to bool and see if they succesfully gets converted.
- **Methods to fix:**
    - I will be using the beam job to resolve the error in which I will simply be setting the invalid `autoplay` to `false`.
    - The safest approach would be to use the conversion function and in the conversion function, after filtering out the invalid video RTE tags which have autoplay as a non boolean value, I will simply mark the value as default, which is False.

      I would like to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Link tags

Link tags contain text and URL attributes, where text is non-empty

^ has 515 errors Frontend validation Backend validation

- **Description of Errors**: Link tag contains `text-with-value` attribute which represents the text for the URL. In some places, we found the value as empty and that is invalid.
- **Why did the error occur in the first place:** Previously we did not have frontend validation so that might be the reason why this error has occurred.
- **Plan to Fix Errors**:
    - The one possible way we can fix this error by simply assigning a default value which can be `URL` or `Link` this way it will be clear to the learner that it represents the link.
    - Another possible solution is to just simply delete the tag the reason being is, as the text value is empty so the link is not visible to the learner so we can simply

remove the link and we are not sure of what to put as a value if not delete it, we cannot take guess here.

- **Methods to fix:**
    - I will be using the beam job to resolve the error in which I will simply be deleting the tag.
    - The safest approach would be to use the conversion function and in the conversion function, after filtering out the invalid link RTE tags which have empty text value, I will simply be deleting the tag.

      I would like to go ahead with the second approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

URL should start with http or https
^ has errors Frontend validation Backend validation

- **Description of Errors**: At some places inside the `url-with-value` attribute, we have values or urls that do not start with `http` or `https`.
- **Why did the error occur in the first place:** we do not have frontend validation so that might be the reason why this error has occurred.
- **Plans to fix:** If the URL starts with `http` we are planning to replace it with `https` and if other than this we simply plan to delete the tag.
- **Methods to fix:** Planning to go ahead with the conversion function approach, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

Note - Currently Harsh(Lawfull2002) is working upon this issue and you can find his PR here. I will provide my assistance where needed.

text-with-value attribute should be present in the link tag
^ has 313 errors Frontend validation Backend validation

- **Description of Errors:** `text-with-value` attribute is not present inside the link tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** I'm simply planning to remove the tag.
- **Methods to fix:**
    - One approach would be to use the beam job to resolve the error.
    - I'm planning to use the conversion function to resolve the issue, I will simply remove the tag.

I would like to go ahead with the second method which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

url-with-value attribute should be present in the link tag
^ has 1 errors Frontend validation Backend validation

- **Description of Errors:** `url-with-value` attribute is not present inside the link tag.
- **Why did the error occur in the first place:** Not sure on how this may have occured because even if the field is empty we used to store the empty strings.
- **Plans to Fix Errors:** I'm simply planning to remove the tag.
- **Methods to fix:**
    - One approach would be to use the beam job to resolve the error.
    - Another approach would be to simply go ahead and do it manually.
    - I'm planning to use the conversion function to resolve the issue, I will simply remove the tag.

  I would like to go ahead with the third approach which is to go ahead with the conversion function, the reason would be even if the user reverts back to the invalid lessons the data will still pass through the conversion function and it will become valid again. After writing the conversion function a migration job will be run which will edit the current datastore data.

NOTE: You can find all the decisions accumulated here in this sheet.

Step 3: Move and fix data in Google Cloud Storage

This section includes 3 parts which are as follows -

- Introduction to GCS IO for Beam jobs, which will allow Beam jobs to work with files in GCS.
- Validate that existing files in GCS have the correct MIME types (#13480), and fixing those types if needed.
- Migrate profile images from UserSettingsModel to GCS and also generate WebP for profile images (does not include frontend changes)

Third-Party Libraries

N/A

“Service” Dependencies

N/A

Impact on Other Oppia Teams

All the teams which work upon storage models will be impacted and this will be a good add-on for the future.

Key High-Level and Architectural Decisions

N/A

Risks and mitigations

Potential Risk	**Mitigation**
Reliability risk - As I will be working with the conversion function we will indirectly be working with the datastore and it includes some potential risk, like when we remove any answer group from the exploration and that may lead to state disconnection.	I think after writing the conversion function I will have one migration audit job that can be run before the actual migration job. I can run that audit job several times in case some changes are suggested as this audit job does not make any changes to the datastore. Please note that to prevent the potential risk that I have mentioned in the `Potential Risk` section, I have already ran an audit job to check for the state disconnection and have planned the methods to prevent them. So, with this I can simply avoid the state disconnection risk that we have. After all the changes are done I will simply go ahead and request for the actual migration job. When we run audit job and migration job we also run validations on the explorations, that will help us to detect any irregularities in the audit job and that way we can prevent any critical changes.

Implementation Approach - Milestone 1

I will be using conversion functions to handle all the errored datas that is reported above. As all of the errored data is related to exploration I will be adding the conversion function to `exp_domain.py`. Currently the latest Exploration version is `56` and the current State version is `51`. I will be adding two functions to the file which will be `_convert_v56_dict_to_v57_dict` for the Exploration and for the State it will be `_convert_states_v51_dict_to_v52_dict`.

The Exploration conversion function would look something like this -

```
    @classmethod
    def _convert_v56_dict_to_v57_dict(cls, exploration_dict):
```

```
        """Converts a v56 exploration dict into a v57 exploration dict.
        Version 57 adds few exploration validation checks which are categorized
as General State validation, General Interaction validation and General RTE
validation.

        Args:
            exploration_dict: dict. The dict representation of an exploration
                with schema version v56.

        Returns:
            dict. The dict representation of the Exploration domain object,
            following schema version v57.
        """
        exploration_dict['schema_version'] = 57

        exploration_dict['states'] = cls._convert_states_v51_dict_to_v52_dict(
            exploration_dict['states'])
        exploration_dict['states_schema_version'] = 52

        return exploration_dict
```

The State conversion function will have three functions inside it and those will be for general state, general interactions and general RTE respectively. This way it will be easy to categorize each checks and it will be easy to keep track of.

```
    @classmethod
    def _convert_states_v50_dict_to_v51_dict(cls, states_dict):
        """
        """
        # Update general state validations.
        states_dict = update_general_state(states_dict)

        # Update general state interaction validations.
        states_dict = update_general_state_interaction(states_dict)

        # Update general state RTE validations.
        states_dict = update_general_state_rte(states_dict)

        return states_dict
```

The structure would look something like above and inside all the functions we will be changing the values of states as per the checks.

How things will work?
Please note that I'm planning to create only one PR for all the checks related to conversion function and 3 PR's for all the backend validations.

After completing all the work related to the conversion function and after testing it locally, I will be submiting the request to check this on the backup server to see if everything works as intended and it do not break anything. In this check no data will be changed it will just check if everything works fine or not. After the check passes then we will go ahead and run a migration job which will update the version of each exploration as well as the state to the latest one and will update the datastore.

What happens if creator reverts to the previous version?
When the conversion function will run it will create a new history version for each exploration given the most recent history version. Whenever a previous version of exploration is loaded the code checks the version and if it is not latest then it will pass through the conversion functions and will become the latest. This will ensure that our data never gets invalid again.

We can take a look at `exp_fetchers.get_exploration_by_id()` function which then calls the `get_exploration_from_model()` function where we check the state schema version and if it is not latest then we call `_migrate_states_schema` and from there it handles the updation of the states until it reaches the latest.

Backend Validation
The backend validation of all the checks will be added even though no errors are reported for some checks. The backend validation for the exp state will be added in the `validate()` function of `Exploration` model in the `exp_domain.py` file.
The checks which are related to the interactions will be placed in `InteractionInstance.validate()` function.
The checks related to the RTE will be placed inside the `SubtitledHtml.validate()` function present in `state_domain.py` because the checks will be valid for both the Questions and Explorations.

Please note that there is only one check that should only be present in the `curated` exps which is `Image should have an SVG extension for curated exps` and this check will be present in the `validate_exploration_for_story()` function in which checks specific to curated explorations are present. This function is called when we edit our exploration or when we go ahead and add our exploration to the chapter.

Steps for Milestone - 1

1. AUDIT STEP
    a. Create PRs for audit jobs to detect irregularities in data
    b. Run “detect irregularities” audit jobs on backup server to get list of errored data
2. Finalize PR for entity conversion code
3. Finalize PRs for backend validations
4. MIGRATION STEP on backup server

    a. Deploy entity conversion PR to server
    b. Run the audit job, run the code without committing any changes to the datastore. Repeat this step until no errors reported
    c. Run the entity migration job from the PR.
    d. Re-run “detect irregularities” audit jobs on server to confirm all data is fixed
5. MERGE STEP
    a. If all data is fixed on backup server, then can merge entity conversion PR and backend validation PRs into oppia/develop
6. MIGRATION STEP on test server
    a. Deploy entity conversion PR to server
    b. Run the audit job, run the code without commiting any changes to the datastore. Repeat this step until no errors reported
    c. Run the entity migration job from the PR until it passes.
    d. Re-run “detect irregularities” audit jobs on server to confirm all data is fixed
7. MIGRATION STEP on production server
    a. Deploy entity conversion PR to server
    b. Run the audit job, run the code without commiting any changes to the datastore. Repeat this step until no errors reported
    c. Run the entity migration job from the PR until it passes. Deploy the backend validations to production as soon as the migration job is done.
    d. Re-run “detect irregularities” audit jobs on server to confirm all data is fixed

Implementation Approach - Milestone 2

It includes 2 parts which are as follows -

- Introducing GCS IO capability for Beam jobs, which will allow Beam jobs to work with files in GCS.
- Migrate profile images from UserSettingsModel to GCS and also generating WebP for profile images (does not include frontend changes)

Introduction to GCS IO for beam jobs

For introduction of GCS IO to the beam jobs I'm planning to use `apache_beam.io.gcp.gcsio` module, you can find the documentation here. The reason why I'm not using the `apache_beam.io.gcp.gcsfilesystem` is because under the hood this module is using the `gcsio` module only, for properties that we are planning to implement such as `open`, `write`, `delete` it is using `gcsio`.
I will be creating a new file named `gcs_io.py` inside `core/jobs/io`, which will include the following properties -

- Reading from files
- Writing into files
- Modifying file metadata
- Deleting files
- Getting the list of files in a folder

For this we should know some key terms that google uses, such as `Organization`, `Project`, `Bucket` and `Object`.

- **Organization**: Your company, called Example Inc., creates a Google Cloud organization called exampleinc.org.
- **Project**: Example Inc. is building several applications, and each one is associated with a project. Each project has its own set of Cloud Storage APIs, as well as other resources.
- **Bucket**: Each project can contain multiple buckets, which are containers to store your objects. For example, you might create a photos bucket for all the image files your app generates and a separate videos bucket.
- **Object**: An individual file, such as an image called puppy.png.

We will have in total of 5 classes each inheriting from `beam.PTransform`. The classes are

- ReadFile
- WriteFile
- ModifyFileMimeType
- DeleteFile
- GetFiles

Each class will consist of `expand()` function where the actual logic will go. Our structure would look something like below -

```
class ReadFile(beam.PTransform):
   """Read file data from GCS."""
   def expand(
       self, pbegin: pvalue.PBegin
   ) -> beam.PCollection[datastore_services.Model]:
       """Returns PCollection with file data."""
       return (
           pbegin.pipeline
           | 'Read the file %s' % self.path >> beam.Map(
               self._read_file()
           )
       )


   def _read_file(self, filename):
       """Helper function to read the contents of a file."""

       gcs = io.gcsio.GcsIO()
       return gcs.open(filename, mode='r')

class WriteFile(beam.PTransform):
   """Write file data to GCS."""
   return (
           pbegin.pipeline
           | 'Read the file %s' % self.path >> beam.Map(
               self._write_file()
           )
       )

   def _write_file(self, filename, data):
      """Logic to write into file"""
      gcs = io.gcsio.GcsIO()
      with gcs.open(filename, mode='w') as f:
        f.write(data)

class DeleteFile(beam.PTransform):
   """Delete the file from GCS."""
   gcs = io.gcsio.GcsIO()
   # Will be using gcs.delete()

class ModifyFileMimeType(beam.PTransform):
   """Modify the file MIME type stored in GCS."""
```

```
   # Will be using gcsio.GcsIO.open(mode='w')-
https://beam.apache.org/releases/pydoc/2.2.0/apache_beam.io.gcp.gcsio.html#apac
he_beam.io.gcp.gcsio.GcsIO.open

class GetFiles(beam.PTransform):
   """Get the files present inside the folder."""
   # Will be using gcsio.GCSIO.glob()-
https://beam.apache.org/releases/pydoc/2.2.0/apache_beam.io.gcp.gcsio.html#apac
he_beam.io.gcp.gcsio.GcsIO.glob
```

Testing plan:

I will be using the above functionalities for the next section which is "Validate that existing files in GCS have correct MIME types" but before that I'm planning to do a small test. I will do the following:

- I will create a dummy file and upload to GCS
- I will fetch that file or read the data from the file
- Edit the file or write to the file
- Change the MIME type of the data
- After that I will delete the file

Migrate profile images from UserSettingsModel to GCS and also generate webP for images

Current status

Currently the profile images are stored in `UserSettingsModel` as a base64 string which is not a good practice. We will be migrating all our profile images from the `UserSettingsModel` to GCS. The profile image data are stored as base64 string in `profile_picture_data_url` field of the `UserSettingsModel`.

There are two handlers for retrieving the profile image:

- ProfilePictureHandler — used for retrieving the profile image of the logged user
- ProfilePictureHandlerByUsernameHandler — used for retrieving the profile image of the user with a particular username

If the user did not set up their profile image, they should most probably have Gravatar identicon set up as their image, because gravatar is added for all new users (#1672) and all the old UserSettingModels were migrated (#1778). I will be writing a one off job to verify that the `profile_picture_data_url` field is set.

Expected status

The profile image will be saved in the GCS under path user/<username>/profile_image.png (the file format can be different than png; similar to how exploration images are saved exploration/<exploration_id>/some_image.png). We can also perform one thing which is to convert the base64 string to webP image and then store it to GCS. I will make sure that if the `username` is changed then the appropriate name is also changed for the image. We need not to be worried about the filepath as everything will be present inside the GCS bucket and GCS has no concept of nested folders, it is just for our convinience that we name our files using `/` so that it will be easy to understand.

`ProfilePictureHandler` and `ProfilePictureHandlerByUsernameHandler` handlers can be removed since they would not be needed.

The current existing images will need to be migrated to GCS and then the `profile_picture_data_url field` should be removed from the `UserSettingsModel`.

For newly created users we will generate the gravatar image and upload it to the GCS. When a user uploads a new version of their image it will replace the one that is already in GCS.

The Big Picture

This section will include 3 beam jobs -

1. **Find and fix pictures**: One job will include the filtering of invalid `profile_picture_data_url` field inside the `UserSettingsModel`. In case we find some models that do not have correct `profile_picture_data_url` in models I will be generating the gravatar for it in the another job. Details to this section can be found in the `Step 1` of `Approach` section.

2. **Store images to GCS**: This job will include the conversion of base64 string to both webP and PNG image and then store it to the GCS. I will be using webptools library to generate the webP image. Details to this can be found in the `Step 2` of the `Approach` section.

3. **Modify the frontend and backend:** Modify the frontend and backend accordingly as we will no longer be fetching the base64 string from the `profile_picture_data_url` field. Details to this can be found in the `Step 3` of the `Approach` section.

4. **Cleanup**: This job will simply remove the `profile_picture_data_url` field from the `UserSettingsModel` as we will no longer be requiring it. This will be the part of the `Step 4` in the `Approach` section.

We won't require to introduce any new field to the model as we can simply retrieve the data by this URL - `user/username/profile_picture.png`.

So in total this section will consists of 4 PR's.

Approach

Step 1: Find and fix pictures

This section will consist of 2 Jobs:

1. AuditInvalidProfilePictureJob
2. FixInvalidProfilePictureJob

First of all we will run `AuditInvalidProfilePictureJob` to check if we have any invalid `profile_picture_data_url` field or to be precise we will check if this field is None or an empty string. Other than this, convert the base64 string to image and then check for the dimensions, filter out the images having dimension other than 150x150.

Now assuming that we will be getting some errored data, we will next run the `FixInvalidProfilePictureJob` to fix the profile picture. This job will include generating "gravatar" for the invalid model and I will be using `user_services.get_gravatar_url()` method to generate the profile picture for the users.
For the images with non standard dimension we are planning to manually edit the images and then upload directly to GCS.

Now to verify that we have fixed the invalid models we will be running `AuditInvalidProfilePictureJob` again.

Step 2: Store images in GCS

As we have now introduced GCS IO to beam jobs, I will be using the `WriteFile` to write the files to the GCS. Before doing that I will be converting the base64 string to webp, for this I will be using webptools library. I would require to use BeamJob to perform this.

It would look something like -

```
import webptools

def _convert_base64_to_webp(base64str: str):
   """Convert base64 to webp image."""
   return webptools.base64str2webp_base64str(
       base64str=base64str, image_type="webp", option="-q 80", logging="-v")

base64_to_webp_images = (
   self.pipeline
   | 'Get all user settings model' >> ndb_io.GetModels(
       user_models.UserSettingsModel.get_all(include_deleted=False))
   | 'Map to user profile picture and username' >> beam.Map(
       lambda model: (model['profile_picture_data_url'], model['username']))
   | 'Convert base64 to webp' >> beam.Map(
       lambda model: (self._convert_base64_to_webp(model[0]), model[1]))
   | 'Write file to GCS' >> beam.Map(
       lambda model: gcs_io.WriteFile(
```

```
            model[0], 'user/%s/profile_image.webp' %(model[1]))
    )
)
```

Step 3: Modify the frontend and backend

Frontend changes: Currently to get an image we use `image-preloader.service.ts` file inorder to get the base64 data from the backend and then converting them to image and then we load it. Instead of this we will be using `profile-link-image-backend-api.service.ts` file functions to directly call the GCS via API to get our image. As the file structure in GCS would be something like `user/<username>/profile_image.png` by this we will directly get our image.

Now we will also need to make the changes to the file where we edit our profile picture image and that can be found in the file `edit-profile-picture-modal.component.ts`.

Please note that I will be coordinating with Eric regarding the frontend changes in order to make sure that the complete section works correctly.

Handling the frontend changes in backend: So after the changes will take place we will no longer be saving the base64 string we will directly be storing the image to the GCS and the method would look something like I have mentioned in the `update_profile_picture_data_url` function below. Now we will also need to make changes to the controller layer as the data will be passing through that.

We will be able to fetch the data for the profile from storage in the frontend only, so we need not to perform any changes to send the URL for the retrieval.

I will be editing the `PreferencesHandler` present inside the `core/controllers/profile.py`. Both the `get` and `put` request will be edited as we no longer have to send the URL for the profile to the frontend as it can be fetched directly.

I will remove the `ProfilePictureHandler` and `ProfilePictureHandlerByUsernameHandler` that is present in the `core/controllers/profile.py` as we will no longer be needing it.

We will need to modify the `update_profile_picture_data_url` function to upload the image to GCS, that could look something like -

```
raw = incoming_image
filename = filename_from_frontend
filename_prefix = filename_prefix

try:
    file_format = image_validation_services.validate_image_and_filename(
        raw, filename)
except utils.ValidationError as e:
    raise self.InvalidInputException(e)
```

```
fs = fs_services.GcsFileSystem(entity_type, entity_id)
filepath = '%s/%s' % (filename_prefix, filename)

if fs.isfile(filepath):
   raise self.InvalidInputException(
      'A file with the name %s already exists. Please choose a '
       'different name.' % filename)
image_is_compressible = (
file_format in feconf.COMPRESSIBLE_IMAGE_FORMATS)
fs_services.save_profile_picture_to_gcs(
   filename, entity_type, entity_id, raw, filename_prefix)
```

We are not planning to have another field to keep track of the profile picture, as the URL will be something like `user/username/profile_picture.png`. So please note that as soon as the username is updated we have to update the directory in GCS, so whenever the user changes its username we will be renaming the blob. To rename our file we can do something like this -

```
def rename_blob(bucket_name, blob_name, new_name):
    """Renames a blob."""
    # The ID of your GCS bucket
    # bucket_name = "your-bucket-name"
    # The ID of the GCS object to rename
    # blob_name = "your-object-name"
    # The new ID of the GCS object
    # new_name = "new-object-name"

    storage_client = storage.Client()
    bucket = storage_client.bucket(bucket_name)
    blob = bucket.blob(blob_name)

    new_blob = bucket.rename_blob(blob, new_name)
```

Step 4: Cleanup

Now we can simply remove the `profile_picture_data_url` field from the `UserSettingsModel`, I will write a beam job to remove this field from our models.

Launch Plan

1. Audit and fix invalid profile pictures in UserSettingsModel.
    a. Deploy the PR to the backup server.
    b. Run AuditInvalidProfilePictureJob to audit invalid profile pictures.
    c. Run FixInvalidProfilePictureJob to fix the invalid profile pictures.

    d. Run AuditInvalidProfilePictureJob to check if all the profile pictures are fixed.
    e. After the successful run of all the jobs, merge the PR to develop.
2. Backend and frontend changes to store profile pictures directly to GCS and to handle takeout and wipeout services.
    a. Deploy the combined PR to the backup server.
    b. Run StoreProfilePictureToGCSJob to push the profile pictures to GCS.
    c. Run AuditProfilePictureFromGCSJob to check the images stored on GCS and the images in model are same.
    d. Try to upload a dummy profile picture to any user in order to see if it gets stored to GCS.
    e. After the successful run of all the jobs, merge the PR to develop.

    From now on the images will be directly deployed to GCS.
3. Cut the release with (1) and (2) to production.
    a. Run AuditInvalidProfilePictureJob to audit invalid profile pictures.
    b. Run FixInvalidProfilePictureJob to fix the invalid profile pictures.
    c. Run AuditInvalidProfilePictureJob to check if all the profile pictures are fixed.
    d. Run StoreProfilePictureToGCSJob to push the profile pictures to GCS.
    e. Run AuditProfilePictureFromGCSJob to check the images stored on GCS and the images in model are same.
4. Remove all the occurrences of the profile_picture_data_url field from UserSettingsModel.
    a. Deploy the PR to the backup server.
    b. Run RemoveProfilePictureFieldJob, this will remove the field from the UserSettingsModel.
    c. Run the jobs from step 2, which are StoreProfilePictureToGCSJob and AuditProfilePictureFromGCSJob.
    d. After the successful run of all the jobs, merge the PR to develop.
5. Deploy (4) to production.
    a. Run RemoveProfilePictureFieldJob, this will remove the field from the UserSettingsModel.
    b. Run the jobs from step 2, which are StoreProfilePictureToGCSJob and AuditProfilePictureFromGCSJob.

[Web only] Storage Model Layer Changes

N/A

Domain Objects

N/A

User Flows (Controllers and Services)

N/A

Documentation changes

N/A

Testing Plan

E2e testing plan

#	Test name	Initial setup step	Step	Expectation
1.	Testing conversion function	I will create invalid data by commentin g out the frontend validation	I will create some bad data on the develop branch and after that will move to the branch with the conversion function implementation and then will publish it again and then check.	Expectation is to have all the data become valid.
2.	Backend test file	I will write a test file for each and every job.	After writing the beam job I will write a test file for that.	Test files should be passed with expected outputs.
			I will also check via being a "Release Coordinator"	Everything should work fine while testing everything as a "Release Coordinator"

Feature testing

Does this feature include non-trivial user-facing changes? **NO**

Implementation Plan

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

Milestone 1			
1.1	Write the bulk of the Beam Job that audits all explorations and outputs validation errors	July 9	

1.2	Write a supplemental Beam Job that performs additional investigation into specific validations given the previous output; splitting output by private, public, and curated lessons	July 18	
1.3	Of all of the exploration validations, categorize the checks into ones that should be applied to private, public, and/or curated lessons	July 18	
1.4	Conversion function fixes all the State, Interaction and RTE data	September 12	
1.5	Add backend validation checks for exploration State (6 checks)	September 12	<-- raise PR by
1.6	Add backend validation checks + logic for exploration Interactions (32 checks)	September 12	<-- raise PR by
1.7	Add backend validation checks + logic for exploration RTE components (7 checks)	September 12	<-- raise PR by
Milestone 2			
2.1	Introduction of GCS Io to the beam jobs	November 3	<-- raise PR by
2.2.1	AuditInvalidProfilePictureJob + FixInvalidProfilePicturesJob	November 3	<-- raise PR by
2.2.2	MoveImagesFromNDBToGCS + AuditImagesOnGCS	November 6	<-- raise PR by
2.2.3	Frontend and backend changes to handle the images on GCS	November 10	<-- raise PR by
2.2.4	Cleanup of the image data from NDB	November 11	<-- raise PR by

Future Work

I don't think any extra work would be required in the future but in case we lead to some decisions which need to be implemented I will surely complete them.

NOTE - The below section is not included in the Milestone-2, we decided to cover only 2 sections that are mentioned above. Keeping the below section for future references.

Validate that existing files in GCS have correct MIME types

Problem statement

This section will validate the data that is present inside the GCS. We won't be requiring the validation before storing/updating the data as in our case the `content_type` is reused as the MIME type when we store the data in GCS. For reference you can take a look here where we commit our data to the GCS.
We already have validation for the image part as we only save the images with the following extension to the GCS -

- jpeg
- jpg
- png
- svg
- gif

For the audio part we currently store `audio/mp3` as an accepted MIME type and this needs to be replaced with `audio/mpeg`, for reference you can take a look here.

This section consists of mainly 2 parts -

1. Checking if the MIME type and the content type of the files present in GCS are equal or not.
2. All the audio files present in GCS should be `audio/mpeg`.

For the first part, We will need to check the MIME type and the content type which is a metadata associated with the files that are stored in the GCS. Here we are validating the data that is present inside the GCS. We don't need to validate the incoming data as in our case the `content type` and the MIME type are equal and while storing our data we passed the content type as the MIME type.

For the second section, Currently in our codebase, for the audio files the content type is `audio/mp3` and it is not the correct MIME type, the correct MIME type is `audio/mpeg` and we want to remove all the occurrences of `audio/mp3` from both our codebase and our storage. Need to validate all the audio files that are stored in GCS and fix the content type – but before fixing the existing data inside the GCS, we need to fix the data that is committed to GCS. We also have an issue opened for this here.

Solution

Checking if the MIME type and the content type are equal or not
I'm planning to use the `mimetypes` library with the help of which I will be able to get the MIME type of the file and I will check if it is equal to the `content type` or not.
If they are not equal I will simply make them equal.

All the audio files present in GCS should be `audio/mpeg`
To avoid getting audio files with `audio/mp3` as an MIME type I will be editing all the occurences of `audio/mp3` to `audio/mpeg` so that we can stop the commiting of `audio/mp3` files.

I can think of 2 methods inorder to fix the content type of the existing audios stored in the GCS. Both of the methods include to fetch the data from the GCS then validate them and update them with the correct MIME type which is `audio/mpeg`. Please note that we will only be fetching the audio files.

Method 1
There is no direct way in which we can filter out the files on the basis of MIME types. One method that we can use is `list` method that is part of the object, here object simply refers to the file that we want to put into the bucket. While calling the `list`(you can find this function here) in the arguments we will pass `bucket` and `delimiter`. In the bucket we will send the name of the bucket as well as the delimiter as `audio/mp3` which will list all the objects inside the bucket that has `audio/mp3` in their path and we can simply return it.

Method 2
Another method is to check the file metadata, specifically the 'content-type', So whenever we push our image file to the GCS we pass the `content type` as the MIME type. We can simply get the `content type` of the file and then check if it is `audio/mpeg` or not, if it is not we will simply be changing the type to `audio/mpeg`.

I'm planning to go ahead with Method 2 because this is the more general approach I can think of. I can directly use `gcs_io.GetFiles` to get all the files inside the folder and after that I can map them with the `content-type` and filter out.

The Big Picture
This section will consist of 2 jobs -

1. Filter out the incorrect MIME types: I will write one job to first filter out all the files that do not have correct MIME type in our case we will check for the `content_type`. I'm planning to use `mimetypes` library. I will also filter out the audio files that do not have `audio/mpeg` as content type. This will be a one-off job which will just be used to filter out the data and details to which can be found in the 'Approach' section below.
2. Fix the MIME types: I will make the content type and MIME types equal if they are not and also make the audio files content type to `audio/mpeg`.

Firstly, as I have to filter out the files I can simply use the above approach to perform that and after that I'm planning to use the custom class that I have wrote previously to edit the metadata of a file which is `gcs_io.ModifyFileMetadata`. The details can be found in the `Approach` section below.

Approach

1. For the first job to take place, we can do something like below -

```
self.pipeline
        | 'Get all the files' >> gcs_io.GcsIOGetFilesInsideFolder()
        | 'map with content type' >> beam.Map(
            lambda file: (file.name, file.content_type)
        )
        | 'filter mp3 files' >> beam.Filter(
           self.filter_for_incorrect_mime_types
        )

def filter_for_incorrect_mime_types(self, content_type):
    """
    """
    mime_type = mimetypes.guess_type(file)[0]
    if mime_type != content_type:
      return True
    return False

def filter_incorrect_audio_files(self, content_type):
    """
    """
    if content_type == 'audio/mp3':
      return True
    return False
```

2. For the second job as we now have all the files with incorrect MIME types we can replace them. As we know that what we call as `file` is actually an `object` in GCS terms and it consists of the metadata that we have to edit. The field that we need to replace is the `content_type` present in metadata, can be found here.

```
all_invalid_files_with_incorrect_mime_types
| 'Change content-type metadata of the file'
>> gcs_io.ModifyFileMetadata(metadata_to_edit='content_type', value=
mimetype_of_file)
```

```
all_invalid_files_with_incorrect_audio_files
| 'Fix the content type of the audio files'
>> gcs_io.ModifyFileMetadata(metadata_to_edit='content_type',
value='audio/mpeg')
```