

Section 2: About Your Project

Project Details

Project title	Learner Dashboard Redesign
Project size	Medium (~175 hours)
Why did you choose this project?	The UI of the existing Learner Dashboard has inconsistencies across different viewports, which can cause confusion for learners and degrade the overall user experience. By focusing on making the UI consistent across all viewports, this project has the potential to greatly improve the user experience and make it easier for learners to find what they are

	<p>looking for. This consistency will not only benefit learners but will also lead to a more cohesive and professional look and feel to the application.</p> <p>These things motivate me to take up this project.</p>
--	---

Project Timeframe

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

Coding period	<ul style="list-style-type: none"> I will adhere to the above deadlines.
Planned time commitment	<p>I will be on a summer break from May to July, during this time I will be able to commit ~35 hrs/week. From 26th July onward, I will only be able to commit ~25 hrs/week. I may increase the hours based on the pace/requirements of the project.</p>
What other obligations might you need to work around during the summer?	<p>I will be having classes from 29th July onwards. During this time I will be able to commit less time than usual.</p>

Communication Channels

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

I can commit to sending daily updates to my mentor by email, each day I work during the GSoC period.	<ul style="list-style-type: none"> Yes
In addition to the above: how often, and through which channel(s), do you plan on communicating with your mentor?	<p>I will communicate with mentors twice a week via Google Meet, Google chat, and Zoom and update the mentor on my progress. I remain active on google chat and Emails.</p>

Section 3: Proposal Details

Problem Statement

Link to PRD (or N/A if there isn't one)	Learner Dashboard IA Redesign Project Primer
Target Audience	Learners across the world that are using Oppia as a platform to learn, teach new things, and enhance their skills. But the general target users are between 7 and 14 years old.
Core User Need	<ul style="list-style-type: none">● As a learner, I need the same view of the Learner Dashboard on mobile and Desktop.● As a learner, I need a clear understanding of the Learner dashboard when I switch to devices with smaller screens, rather than scrolling and navigating to search for a particular thing.● Having easy navigation improves the overall learning experience.
What goals do we want the solution to achieve?	<ul style="list-style-type: none">● Make the same UI on both small and large screens.● Prevent the learner to scroll and search unnecessarily by making the same UI on both screens.● Make the Learners easily navigate and interact with the Learner dashboard regardless of the device they are using, overall learners feel comfortable and confident while using the learner dashboard.● Make the learner dashboard accessible more to users with disabilities. Users who rely on assistive technologies, such as screen readers, may struggle to navigate a website that has a different UI across different devices.● Taking the above point into consideration, The aim of this project is to actually redesign the dashboard so that it looks good and is intuitive for learners as a way to gather their "learning tasks" in a single easy-to-access place.

Section 3.1: WHAT

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

Key User Stories and Tasks

#	Title	User Story Description (role, goal, motivation) "As a ..., I need ..., so that ..."	Priority ¹	List of tasks needed to achieve the goal (this is the "User Journey")	Links to mocks / prototypes, and/or PRD sections that spec out additional requirements.
1	A returning learner.	As a returning Learner , I want quickly be able to check where I left off, my progress , and check feedback updates .	Must have	After logging in, the learner visits the learner dashboard.	Mock
				Learners navigate to the Home Tab and check the " Continue where you left off " section.	Mock
				Learner navigate to the " Progress Tab " and check the " In Progress " section.	Mock
				Learner will see an indicator on the profile pic. Learner navigates to the profile-pic drop-down menu and Clicks on feedback updates.	
2	Add or edit goals	As a returning Learner, I want to add or edit goals	Must have	After logging in, Navigate to the Goal Tab.	Mock
				Click on the "Add a Goal button".	Mock
				Add or edit the goal and click on the Save button.	Mocks
3	Invitation on the learner group section.	As a returning Learner, I want to check the learner	Must have	After logging in, Navigate to the Home Tab.	

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

		group's invitation to accept or decline it.		Click on "More Info" . Click on the "Accept button" or "Decline button".	Mock
4	Check the complete topic, lesson save for later and start something new.	As a returning Learner, I want to Check all the completed lessons, lessons saved for later, skills, and topics and start something new.	Must have	After logging in, Navigate to "Goal Tab" and check the completed section of the Goal Tab.	Mock
				Navigate to the "Progress Tab" and check the Finished section of Progress tab.	Mock
				Navigate to the "Home Tab" and check the "Lesson you saved for later" section.	Mocks
				Navigate to the "Home Tab" and search for a particular lesson, Learn any topic from the "Learn something new section" or Click on the "Or Explore All Lessons In Classroom" button.	Mock

Technical Requirements

Additions/Changes to Web Server Endpoint Contracts

#	Endpoint URL	Request type (GET, POST, etc.)	New / Existing	Description of the request/response contract (and, if applicable, how it's different from the previous one)
1.	/feedbackupdateshandler	POST	New	Response data . <ul style="list-style-type: none"> numberOfUnreadThreads threadSummaries
2.	/user_progress_in_explorations_handler	GET	New	<ul style="list-style-type: none"> New call is needed to fetch the checkpoint data of the explorations. I will give the array of exploration IDs. In response, I will receive an array of objects that contain checkpoint data. The objects in the array will be ordered in the same sequence as the exploration IDs that were passed and each object contains visitedCheckpointsCounts and totalCheckpointsCounts. Example response data:

				<pre> LearnerExplorationsProgress[{ visitedCheckpointsCounts:2 , totalCheckpointsCounts:5 }, { visitedCheckpointsCounts:3 , totalCheckpointsCounts:6 }] </pre>
--	--	--	--	--

Calls to Web Server Endpoints

#	Endpoint URL	Request type (GET, POST, etc.)	New/ Existing	Description of why the new call is needed, or why the changes to an existing call is needed
1.	/user_progress_in_explorations_handler	GET	New	<ul style="list-style-type: none"> New call is needed to fetch the checkpoint data of the explorations. I will give the array of exploration IDs. In response, I will receive an array of objects that contain checkpoint data. The objects in the array will be ordered in the same sequence as the exploration IDs that were passed and each object contains visitedCheckpointsCounts and totalCheckpointsCounts. Example response data: <pre> LearnerExplorationsProgress[{ visitedCheckpointsCounts:2 , totalCheckpointsCounts:5 }, { visitedCheckpointsCounts:3 , totalCheckpointsCounts:6 }] </pre>

UI Screens/Components

#	ID	Description of new UI component	i18n required?	Mock/spec links	A11y requirements
1.	Feedback Updates Page component	Feedback updates tab will be shifted to the profile pic drop-down menu to make the UI consistent across all the devices.	Yes		

2	Redesigning of learner dashboard	The Learner dashboard is set to undergo a redesign to ensure a consistent user interface (UI) across all viewports, thereby affording learners a comfortable learning experience.	Yes	Mock	
---	----------------------------------	---	-----	----------------------	--

Data Handling and Privacy

Other Requirements

N/A

Section 3.2: HOW

Existing Status Quo

The current learner dashboard is not user-friendly, and learners have trouble finding all their learning tasks in one convenient place.

The current design of the learner dashboard is not optimized for all viewports resulting in a lack of consistency between the display on small and large screens. As a result, users feel confused and uncomfortable while switching between different devices. This inconsistency in design often leads to users having to scroll and search more than necessary, making it difficult to locate the information they need quickly.

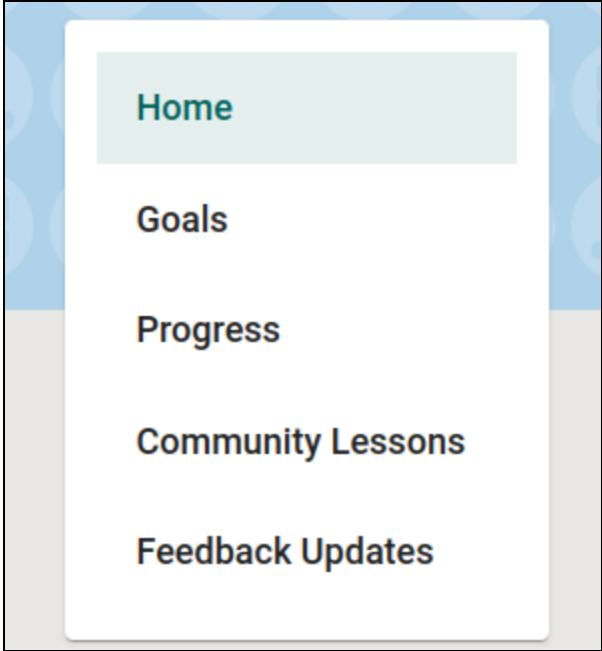
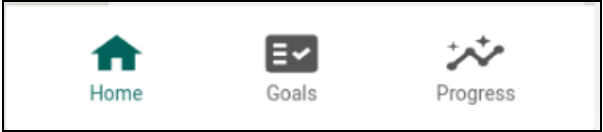
Solution Overview

To improve the design of the dashboard so that it appears visually appealing and is easy for learners to use and to create a user-friendly interface where learners can easily find and access all their learning tasks in one place.

I will take the following steps:

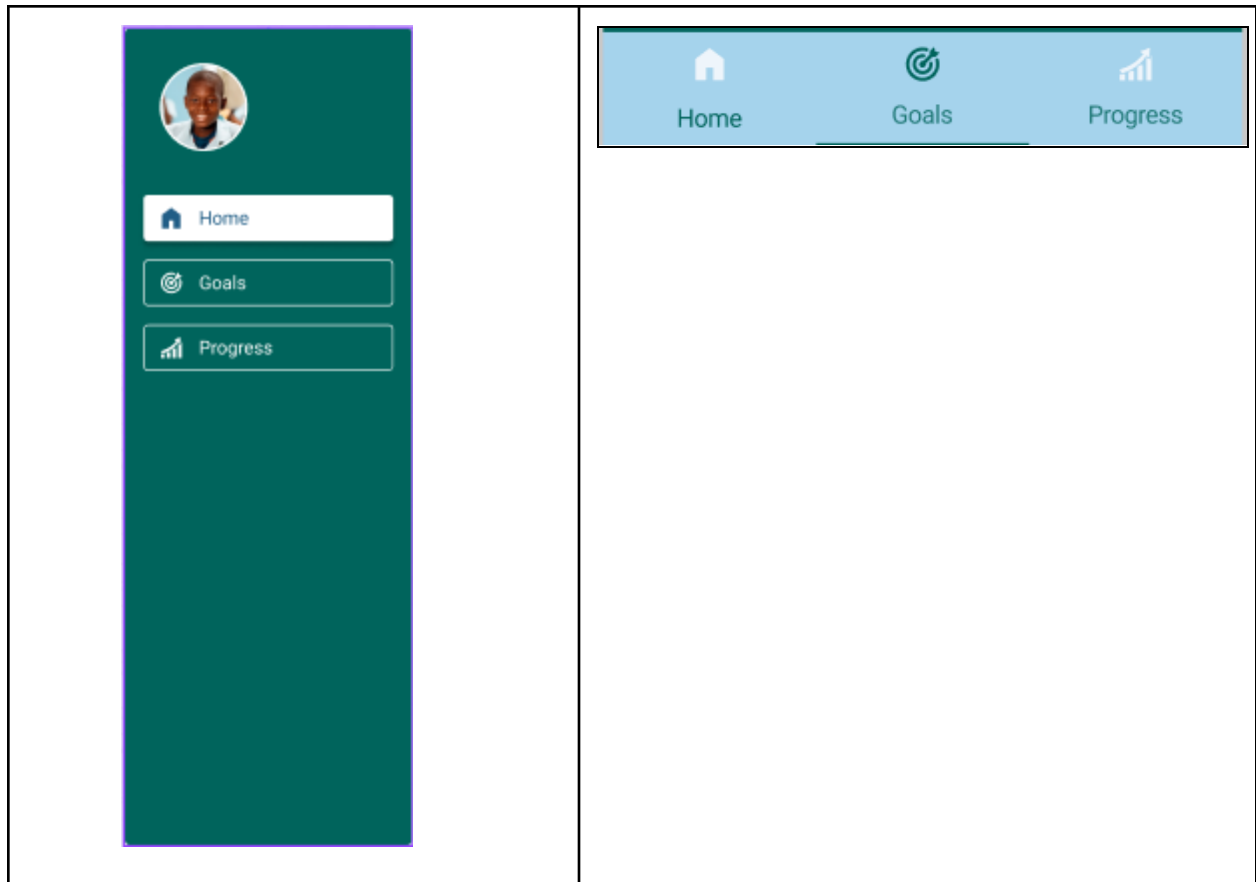
The current Desktop view of the Learner Dashboard contains five tabs, which are Home, Goals, Progress, Community Lessons, and Feedback Updates. However, the new Learner Dashboard has to be designed to include three tabs, namely Home, Goals, and Progress.

→ Tabs in the existing Learner dashboard

Desktop view	Mobile view
	

→After redesigning

Desktop view	Mobile view
--------------	-------------



1. Feedback Updates

In the existing Learner dashboard

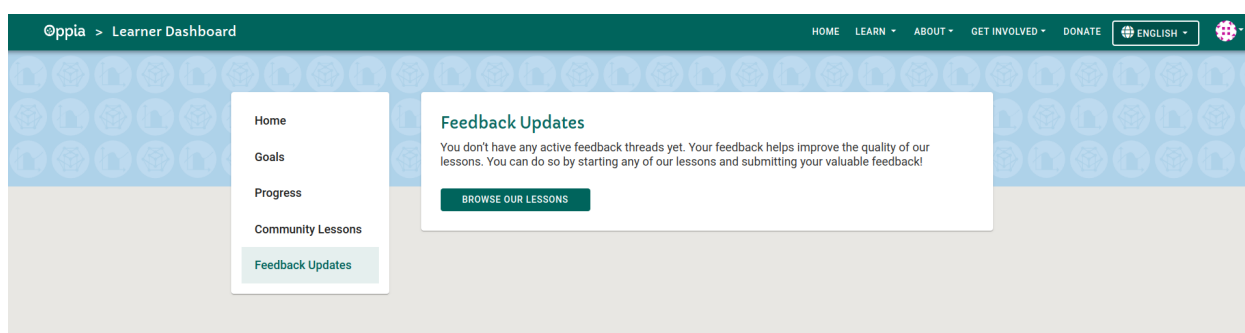
Desktop view

This section comprises the feedback provided by the learner regarding any topic or lesson, alongside the feedback given by contributors from the exploration editor page.

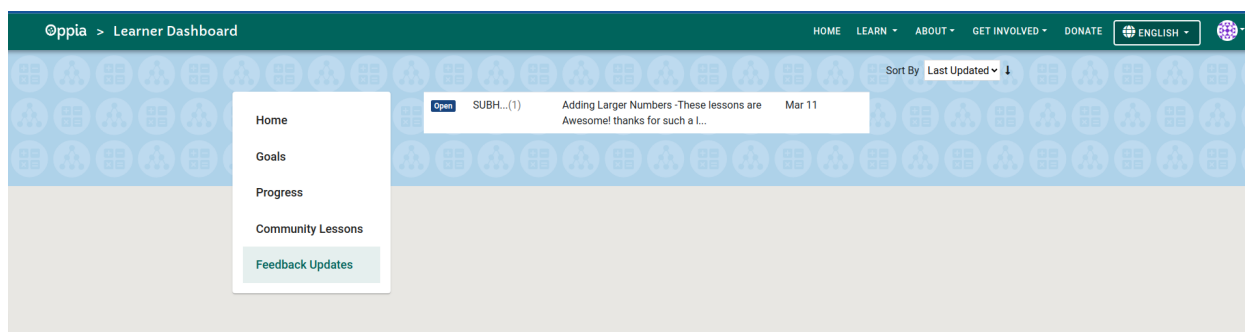
In the absence of any feedback within the feedback updates page, a button is displayed labeled 'BROWSE OUR LESSONS', with text (You don't have any active feedback threads yet. Your feedback helps improve the quality of our lessons. You can do so by starting any of our lessons and submitting your valuable feedback!)

The main navigation bar includes a feedback icon that appears when a learner is studying a lesson/topic or a contributor is on the Creator dashboard. Learners can click on the icon to give feedback, with an option to remain anonymous. Contributors can also give feedback by accessing the feedback tab on the Creator dashboard and starting a feedback thread by entering a subject and message. When learners or contributors provide feedback and revisit the feedback tab, the 'BROWSE OUR LESSONS' button disappears, and a chat box appears containing the lesson/topic name or thread subject, along with the learner's name and the time they provided the feedback. The chat system allows for further communication.

(If no feedback is present – current learner dashboard)



(If feedback is present – current learner dashboard)



Mobile view

The feedback updates are not present in the current mobile view.

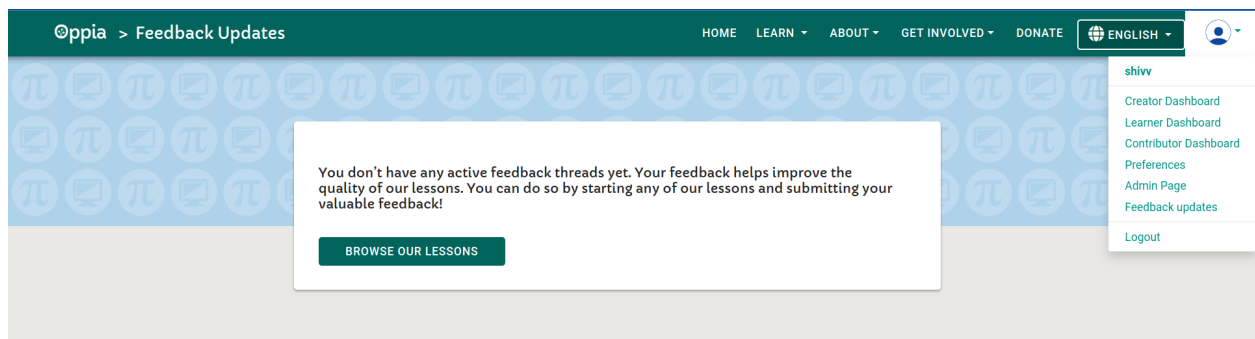
★ After redesigning

Desktop view

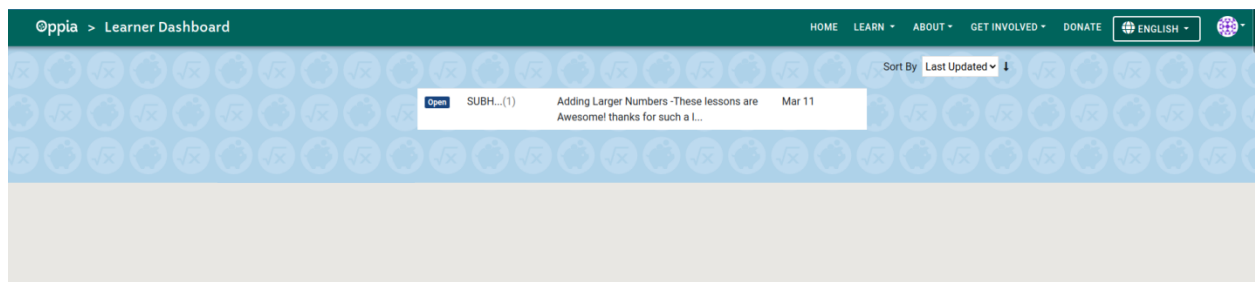
After the redesign, the feedback update is removed from the existing menu and moved to the Profile pic drop-down menu just above the 'Logout' and all the functions present in the existing one remain the same.

For accessing the newly shifted feedback updates, the user has to move to the profile pic drop-down menu and then click the feedback updates. When the user clicks on the “feedback updates” the URL will change to /feedback-updates and a page will open with the title on the navbar “Feedback Updates” containing the two cases given below.

(IF no feedback)



(If feedbacks are present)

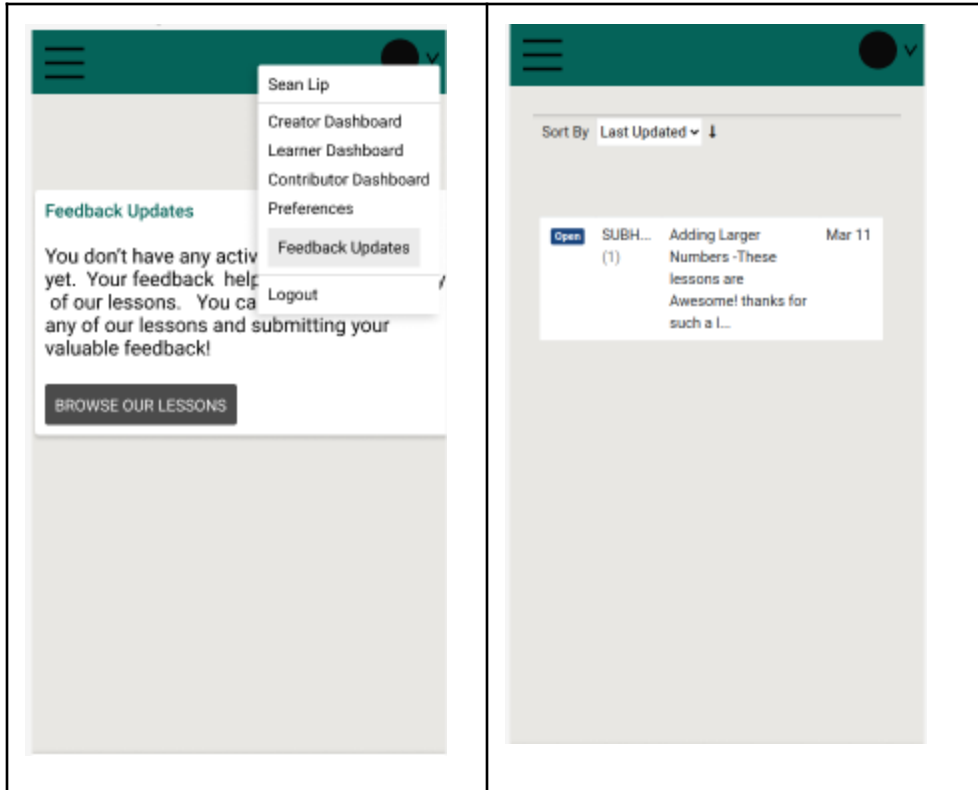


Mobile view

This feature is not present in mobile view currently but after the redesign, feedback update comes in the profile drop-down menu same as the Desktop view.

(IF feedback not present)

(IF feedback present)



2. Home Tab:

In the existing Learner dashboard

In the **Desktop view**, The Home tab is the uppermost tab in the menu, and in the **Mobile view**, it is located in the mobile device's footer, specifically positioned in the leftmost area of the footer. it is the tab that opens by default, greeting the user by name. The Home tab contains various sections, such as *“Continue where you left off”* and *“Learn something new”*.

→ ***“Continue where you left off and Learn something new”***

The cards in the existing “**Continue where you left**” section are the stories suggested by Oppia based on the users’ selected topics in the “Edit Goal” section of the Goals tab. Each story is associated with the next incomplete chapter. To ensure a seamless user experience, the cards in this section are ordered by most recently played. It also contains the lessons (from Classroom or community library) that are in progress. It also contains the next suggested lesson to complete if the learner has joined any learner groups.

The cards in the existing “**Learn something new**” section are the topics and these are topics the user has not yet started, completed, or selected as the user’s current goal in the Edit Goals section. These cards represent areas of knowledge that the user has not yet explored or mastered, and they are waiting to be learned. And it consists of a single row that contains four cards, starting three cards in the row are the topics, and the last card in the row is linked to the Oppia’s math classroom and its color is grey. If there are fewer topics to display in the row (less than 3) then the card that is linked to the classroom gets shifted toward the left.

If the learner reaches the goal limit (5 Goals max) then the suggested topics get removed and replaced by the “BROWSE OUR LESSONS” button with the text “ It looks like you reached the goal selection limit. Head over to the library and explore more explorations”.

The screenshot displays the Oppia Learner Dashboard. The top navigation bar includes links for HOME, LEARN, ABOUT, GET INVOLVED, and DONATE, along with a language selector set to ENGLISH. A sidebar on the left contains navigation options: Home, Goals, Progress, Community Lessons, and Feedback Updates. The main content area features a personalized greeting, "Good Afternoon, jsrk!".

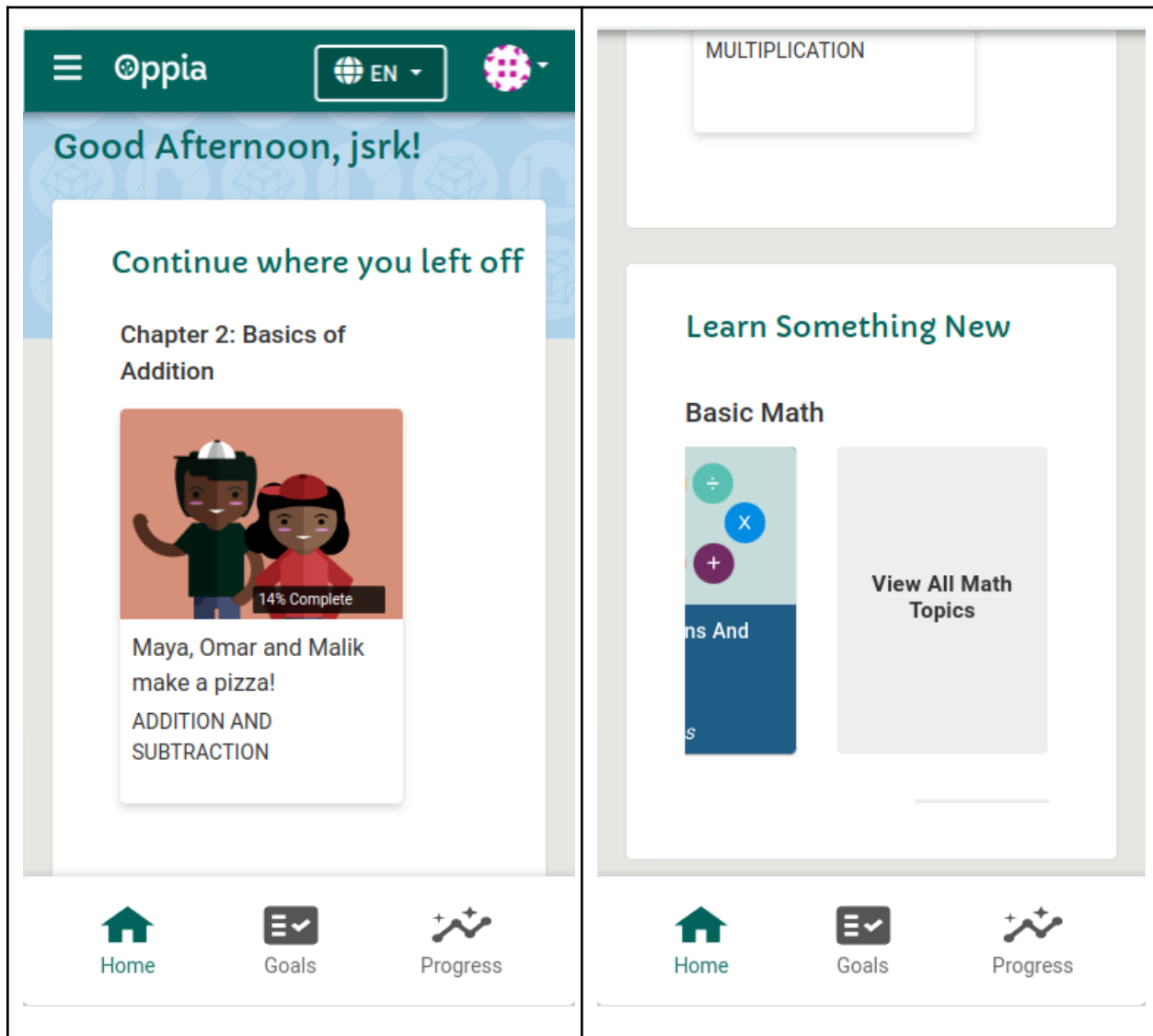
The "Continue where you left off" section shows two cards:

- Chapter 2: Basics of Addition**: A card with an illustration of two children, Maya and Omar, with the text "Maya, Omar and Malik make a pizza! ADDITION AND SUBTRACTION". It indicates "100% Complete".
- Chapter 1: Parts of Multiplication Expressions**: A card with an illustration of a boy and a girl, with the text "Aria wants to plant a garden MULTIPLICATION". It indicates "0% Complete".

The "Learn Something New" section displays four topic cards under the heading "Basic Math":

- Place Values**: A card showing a place value chart with digits 2, 5, 0, 6 in the thousands, hundreds, tens, and ones places respectively, and the numbers 500 and 2000.
- Division**: A card showing a division problem with a dividend of 12, a divisor of 3, and a quotient of 4.
- Expressions And Equations**: A card showing a balance scale with weights and mathematical symbols.
- View All Math Topics**: A grey button to explore more topics.

(Desktop view of current learner dashboard)



(mobile view of current learner dashboard)

→ “Learner Groups”

Learner Groups is a separate tab and a learner group is a collection of learners who are studying the same content or are working towards a common learning goal.

In this Facilitator invites the learner to join groups with other learners. Learners have the choice to accept or decline the invitation. Once the learner accepts the invitation that is the learner becomes a member of a group then the learner will be able to see the learner group syllabus

and choose to share their progress with the facilitator also learner can change sharing permission at any time.

In the **mobile view**, this is a separate tab in the footer of the mobile.

Desktop view	Mobile view
<div><h3>Learner Groups</h3><p>Facilitators can invite you to join groups with other learners. Once you become a member of a group, you will be able to see the learner group syllabus. You can also choose to share your progress with the facilitator. You can change your sharing permissions at any time.</p></div> <div><h4>Your Invitations 1</h4><div><div>group of facilitator name "test2" Facilitator: test2</div><div>View Details✓✗</div></div></div> <div><h4>Your Groups</h4><div><div>Group of facilitator name JSRK Facilitator: JSRK</div><div>View Details</div></div><div><div>Group of facilitator name test1 Facilitator: test1</div><div>View Details</div></div></div>	<div><h3>Learner Groups</h3><p>Facilitators can invite you to join groups with other learners. Once you become a member of a group, you will be able to see the learner group syllabus. You can also choose to share your progress with the facilitator. You can change your sharing permissions at any time.</p></div> <div><h4>Your Invitations 1</h4><div><div>group of facilitator name "test2" Facilitator: test2</div><div>View Details✓✗</div></div></div> <div><h4>Your Groups</h4><div><div>Group of facilitator name JSRK Facilitator: JSRK</div><div>View Details</div></div><div><div>Group of facilitator name JSRK Facilitator: JSRK</div><div>View Details</div></div></div> <div>Dev Mode</div>

★ After redesigning the Home Tab

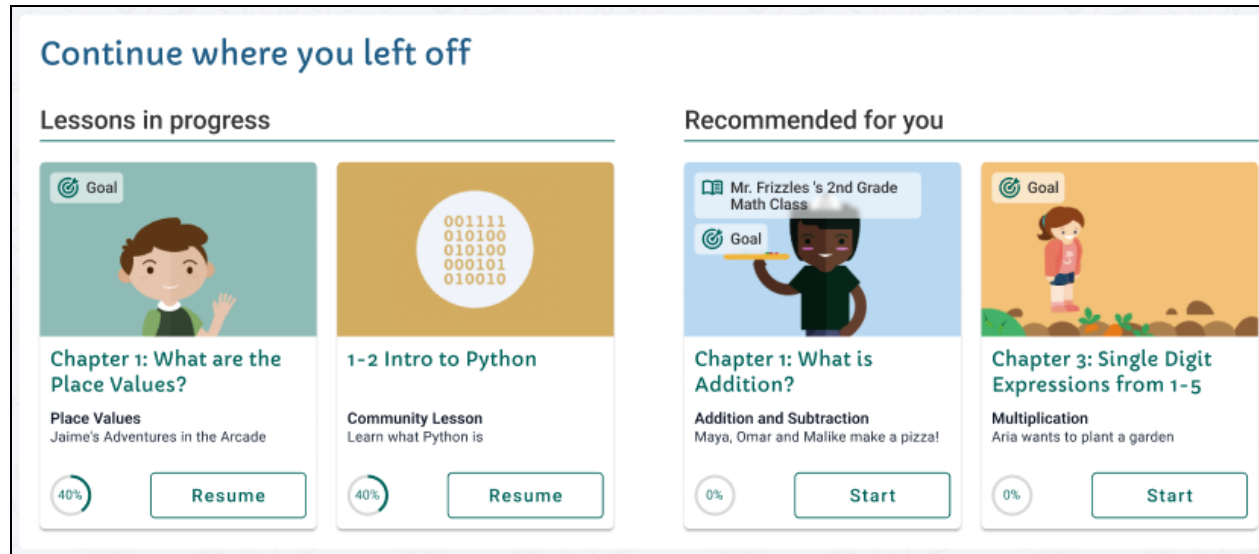
In **Desktop view**, It is updated and the sections in the Home tab will be “Continue where you left off”, “Learn something new”, “Review what you have learned”, and” Learner Groups”.

→ Continue where you left off

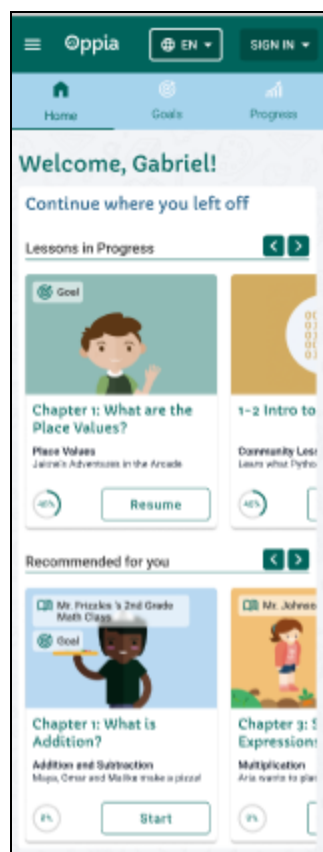
Functionality for this section will remain the same as in the existing one.

The difference is the lesson will get separated into two different sections **InProgress**(lessons in progress) and **Recommended for you**(lessons that are recommended to learn)

Also change in UI as shown



(Desktop view)



(Mobile view)

→ Learn something new

The cards in this section are the topics and these are topics the user has not yet started, completed, or selected as the user's current goal in the Edit Goals section. These cards represent areas of knowledge that the user has not yet explored or mastered, and they are waiting to be learned. This section will contain a search interface enabling the learners to retrieve the information they desire. The search mechanism shall be optimized by allowing the selection of specific categories and languages for refined results. Learners can start any topic by clicking the card and it will take the learner to the specific topic. In case the learners intend to conduct further research on the mathematics courses, they may opt to click on the **"Or Browse all Math Lessons in the Classroom page"** button, which will redirect them to the Math Classroom page.

Learn something new

All Categories ▾ All Languages ▾ Q

Topics available in Oppia's Classroom

Place Values
5 Lessons

Addition and Subtraction
5 Lessons

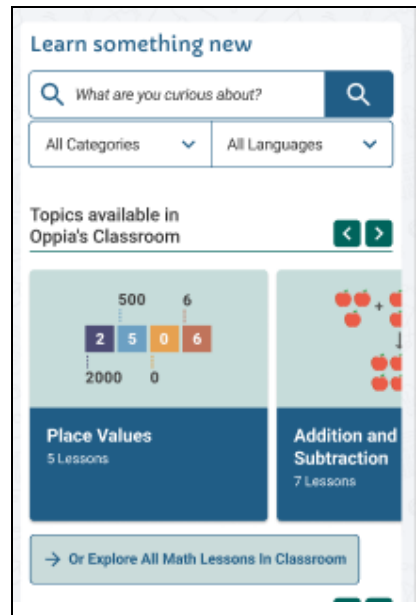
Multiplication
7 Lessons

Division
5 Lessons

Expression Equations
17 Lessons

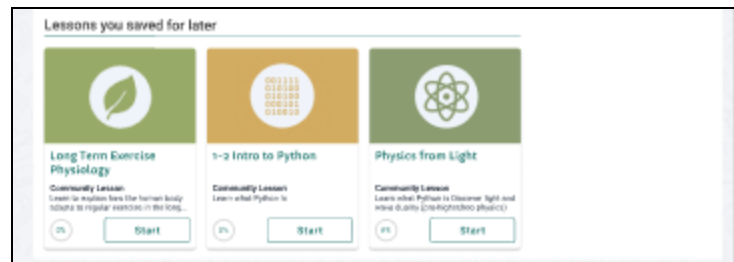
[→ Or Browse all Math Lessons in the Classroom page](#)

(Desktop view)

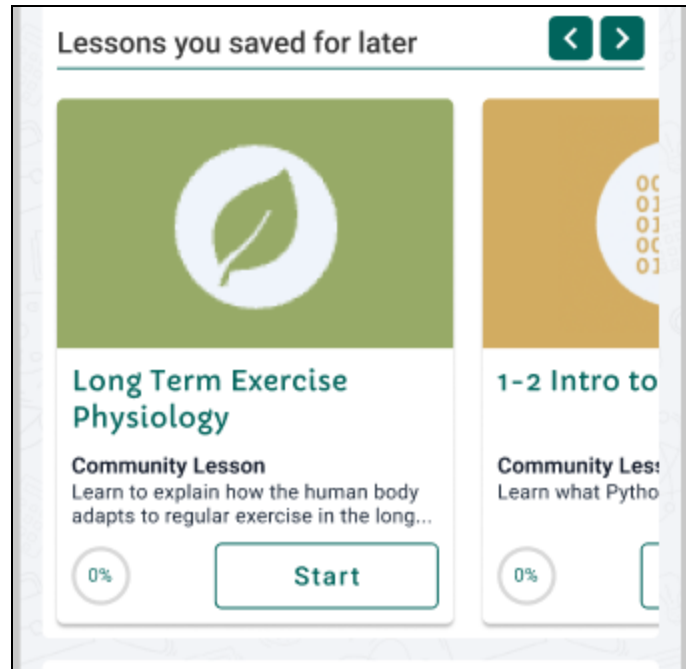


→ “Lessons you saved for later”

This section contains the lessons that are marked by learned to learn later.



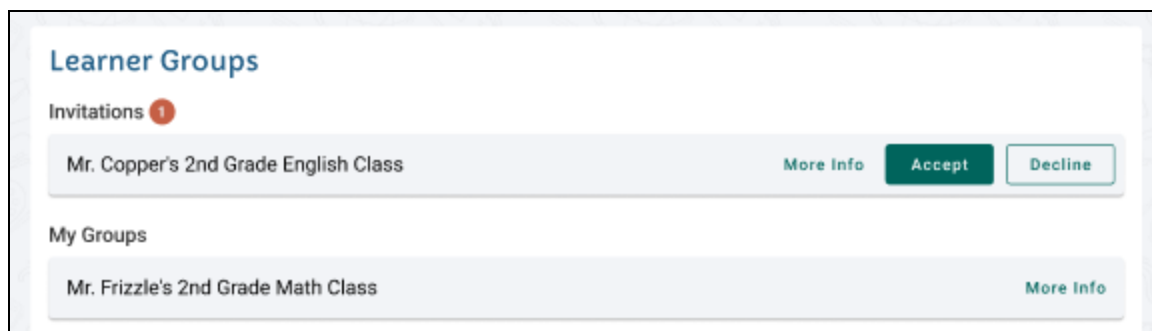
(Desktop view)



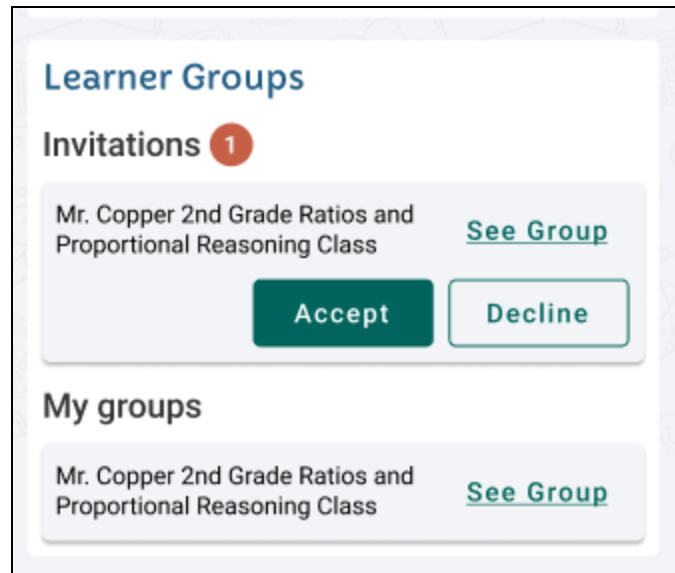
(Mobile view)

→ "Learner Group"

The functionality will remain the same as in the existing one only little bit of UI will be changed.



(Desktop view)



(Mobile view)

3. Goals Tab

In the existing Learner dashboard

In the **Desktop view**, The Goals tab within the application offers a robust mechanism for learners to establish particular objectives and track their advancement toward them and it is the second tab after the Home tab. In **the Mobile view**, the Goals tab is present next to the home tab in the footer of the mobile.

The goals tab consists of three distinct sections, which are **“Edit Goals”**, **Current Goals”** and **“Completed Goals”**

→ Current Goals:

This section contains the list of topics that the learner has selected from the *‘Edit Goals’* section. These goals represent the things that the learner wants to achieve in the near future. By clear understanding of the goals and writing them down, can improve the learner's focus and motivation.

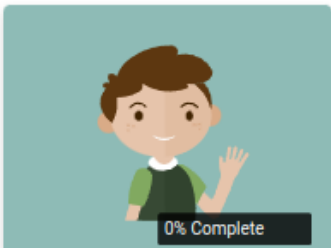
Learners can modify their goals as needed, including deleting existing goals by clicking the delete button next to each goal. In the current dashboard, the delete button only appears for each goal when the user clicks on the first goal's triple dots.

In the existing one **Mobile view** has the same view as the Desktop view.

Current Goals

▼ Learn Place Values

Complete a chapter in 'Place Values'



0% Complete

What are the Place Values?

JAIME'S ADVENTURES IN THE ARCADE

▶ Learn Addition and Subtraction

Complete a chapter in 'Addition and Subtraction'

▶ Learn Multiplication

Complete a chapter in 'Multiplication'

→ Edit Goals(Set a Goal):

In the **Desktop view**, This section displays a list of topics, with a checkbox next to each one. Learners can use this section to set goals by selecting a topic and checking the corresponding box. Once a goal is selected, it will appear in the 'Current Goals' section.

In the existing one **Mobile view** has the same view as the Desktop view.

Edit Goals

Limit of 5 goals



Learn

Topic

Place Values



Addition and Subtraction



Multiplication



Division



Expressions and Equations



Fractions



Ratios and Proportional
Reasoning



Percentages



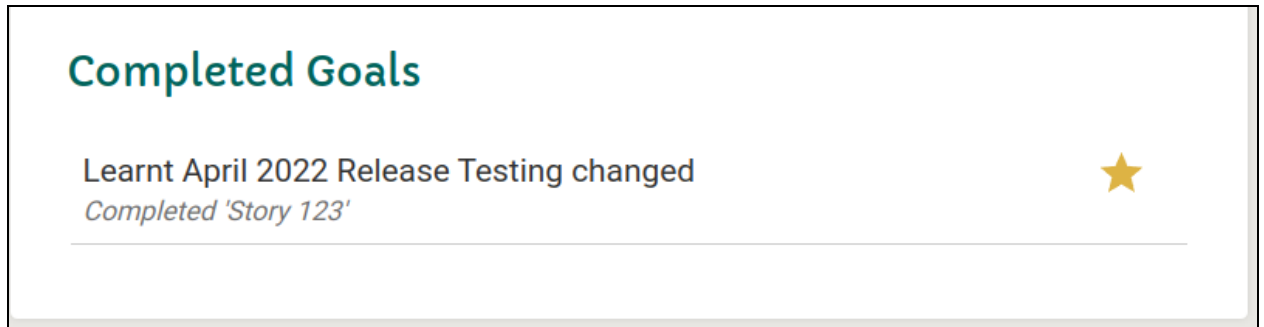
Decimals



→ Completed Goals:

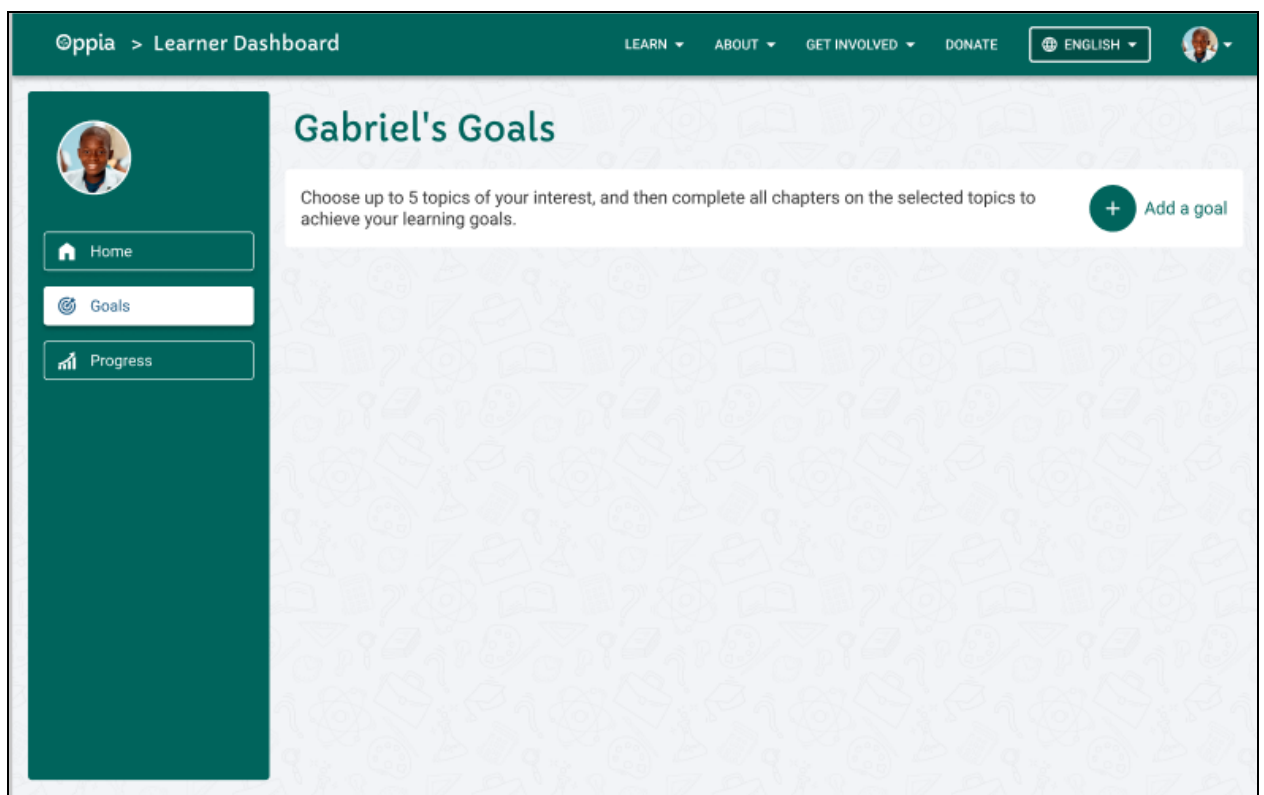
In the **Desktop view**, This section displays the user's achievements, which is a list of all the Goals they have achieved.

In the existing one **Mobile view** has the same view as the Desktop view

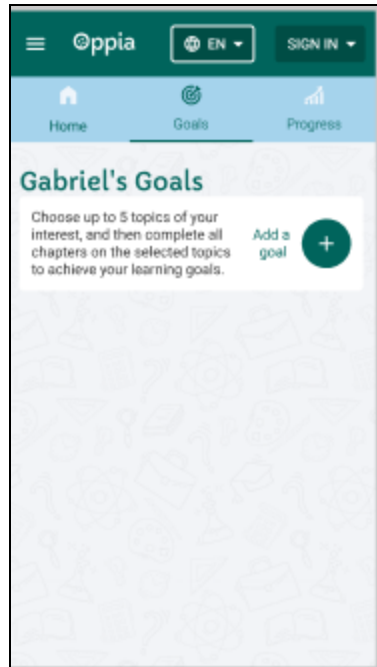


★ After redesigning the Goals Tab

If the user hasn't chosen any goals, they will see an empty goals tab with a plus button (Add a goal) and the text "Choose up to 5 topics of your interest, and then complete all chapters on the selected topics to achieve your learning goals".



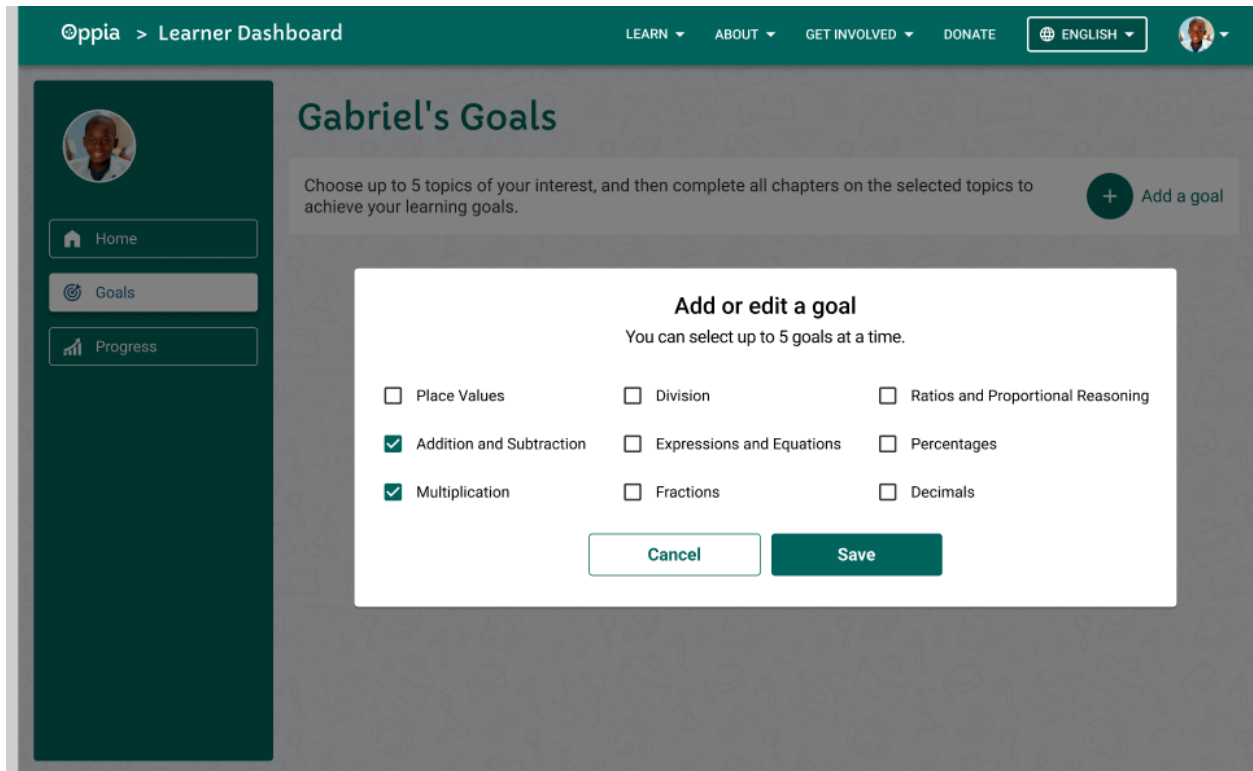
(Desktop view)



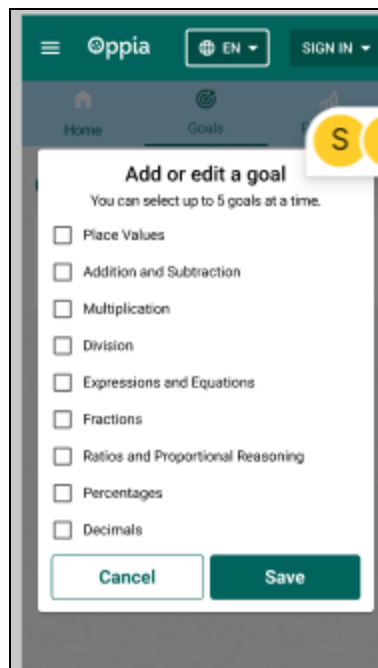
(Mobile view)

→ Add a Goal

In the **Desktop view**, a "Add a goal" button is present on the rightmost corner of the current goals section. Upon clicking the button, a modal window will be displayed that allows the learner to perform multiple actions such as setting new goals and deleting existing ones. It should be noted that the system will allow a maximum of five goals to be set.



(Desktop view)

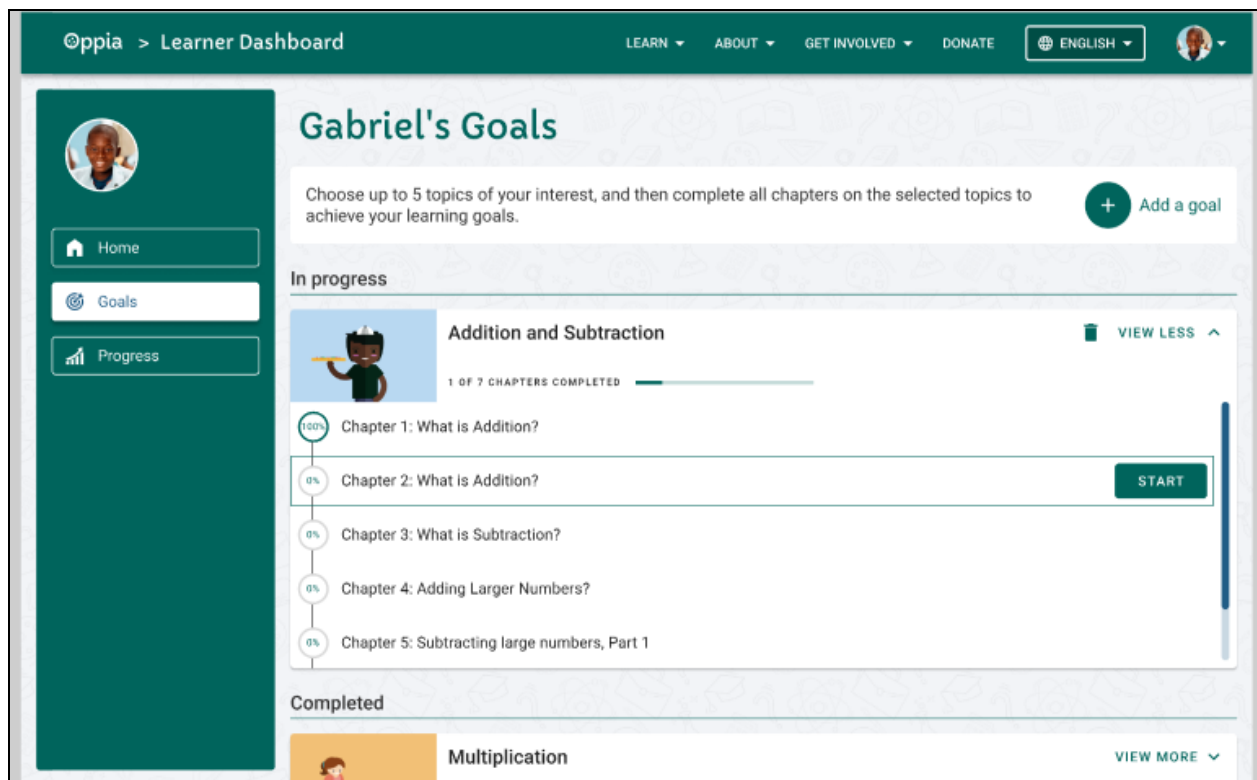


(Mobile view)

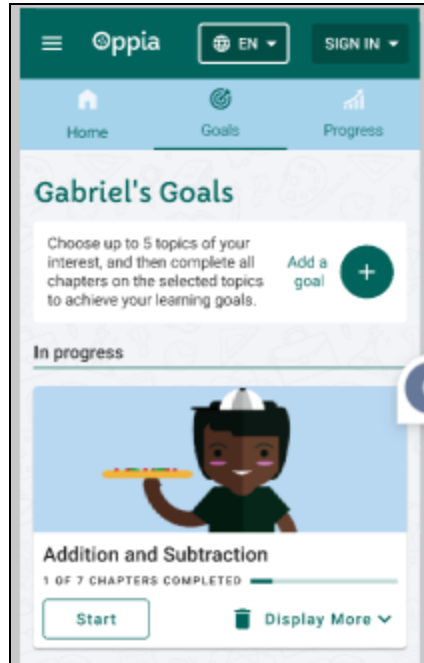
→**In progress:**

After redesign, the “Current goals” section changes to “In progress”.

In the **Desktop view**, there is a section that displays the list of topics that the learner has selected from the '**Add a Goals**' section. The learner can continue their learning from the last left point, and they can view the completion progress of each chapter. When the learner expands a topic by clicking on the drop-down button, all the chapters within that topic will be displayed, with a call-to-action (CTA) button to start any chapter.



(Desktop view)





(Mobile view)

→ **Completed Goals:**

In the **Desktop view**, This section displays the user's achievements, which is a list of all the Goals they have achieved.

Oppia > Learner Dashboard

LEARN ▾ABOUT ▾GET INVOLVED ▾DONATEENGLISH ▾



Home

Goals


Progress

Gabriel's Goals

Choose up to 5 topics of your interest, and then complete all chapters on the selected topics to achieve your learning goals.

+ Add a goal

Completed



Multiplication

VIEW LESS ^

7 OF 7 CHAPTERS COMPLETED

100% Chapter 1: Lorem ipsum

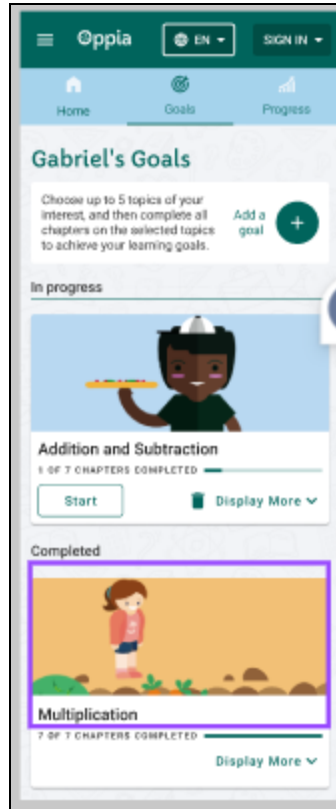
100% Chapter 2: Lorem ipsum

100% Chapter 3: Lorem ipsum

100% Chapter 4: Lorem ipsum

100% Chapter 5: Lorem ipsum

(Desktop view)



(Mobile view)

4. Progress Tab

The Progress tab is a highly valuable feature that enables the learner to closely track their progress towards their learning objectives.

In the existing Learner Dashboard

In the **Desktop view**, This tab is divided into two sections '***Skill Proficiency***,' '***Stories Completed***'

→ '***Skill Proficiency***':

This section facilitates skill mastery through deliberate practice. When learners define their objectives in the 'Edit Goals' section of the 'Goals Tab,' the associated topic will automatically populate in the skill proficiency section, together with the corresponding subtopics.

The progress of the overall topic is shown by a circle (percentage) and the progress of each subtopic is shown by a percentage, in front of each subtopic.

It consists of an 'i-icon'(i.e information icon) that gives the information of the badge that the user gets based on the progress in the overall topic percentage (i.e average percentage of all subtopics).

Here is the info:

Overall topic percentage(%)	Badge
Less than 60	Empty badge
60 to 79	Bronze
80 to 99	Silver
100	Gold

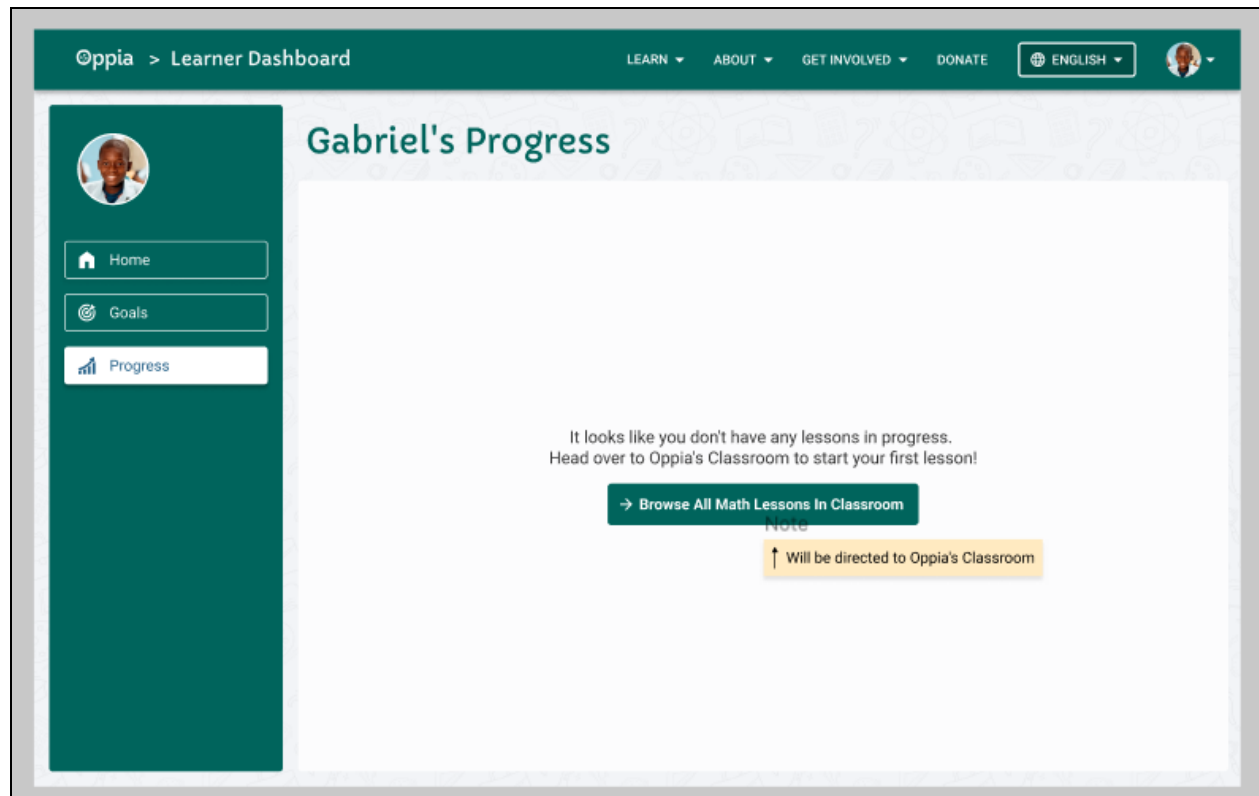
→ **"Stories completed":**

This section contains all the stories completed by the learner. If the learner had not completed any stories then this section is not visible.

The "Lessons" section can be found in the Mobile view, alongside the two aforementioned sections. This section is equivalent to the community tab in the current learner dashboard, as it includes lessons that are currently ongoing (incomplete), finished, and all of them combined.

★ **After redesigning the Progress Tab**

When the user clicks on the Progress Tab and if the user has not started any lessons then it shows a button(Browse all math lessons in Classroom) with the text given below in the image.



(Desktop view)



(Mobile view)

The “**In Progress Tab**” will contain the three sections


→ **In Progress:**


This section will contain the lessons that are in progress from the classroom and community.

Oppia > Learner Dashboard

LEARN > ABOUT > GET INVOLVED > DONATE

ENGLISH





Home

Goals

Progress


Gabriel's Progress

In Progress

Classroom Lessons

Mr. Johnson's 2nd Grade Class

Goal



Chapter 1: What are the Place Values?

Place Values


Jaime's Adventures in the Arcade

40%

Resume

Mr. Johnson's 2nd Grade Class

Goal



Chapter 3: Single Digit Expressions from 1 - 5

Multiplication


Aria wants to plant a garden

40%

Resume

Mr. Johnson's 2nd Grade Class

Goal



Chapter 1: What is a Decimal?

Decimal


Reena's Race to the Finish

40%

Resume

Mr. Johnson's 2nd Grade Class

Goal



Chapter 4: Finding a Percentage of a Number

Percentages

Monica Helps at the Jewelry Store

40%

Resume

Display More

Community Lessons

001111
010100
010100
000101
010010


1-2 Intro to Python

Community Lesson

Learn what Python is

40%

Resume




Physics from Light

Community Lesson

Learn what Physics is Discover light and wave duality (pre-highschool physics)

40%

Resume



Long Term Exercise Physiology

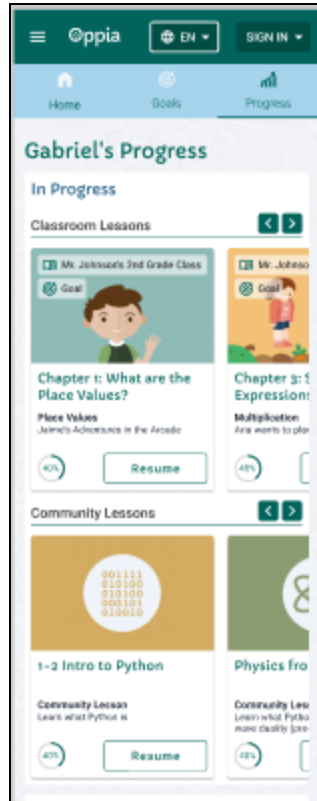
Community Lesson

Learn to explain how the human body adapts to regular exercise in the long...

40%

Resume

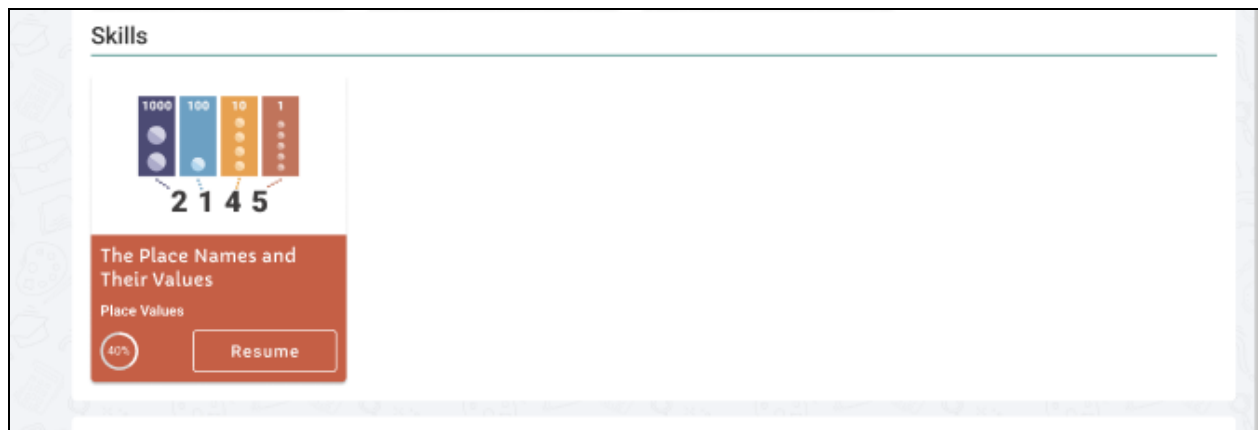
(Desktop view)



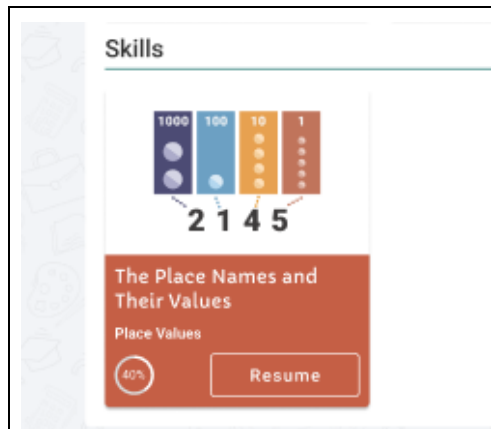
(Mobile view)

→ **Skills:**

This section has the same functionality as the existing one, but differs in UI.



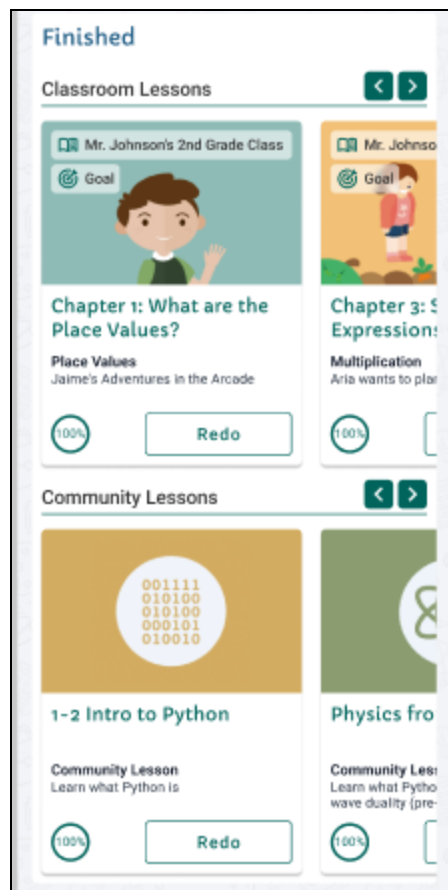
(Desktop view)



(Mobile view)

→ **Finished**

This section will contain all the Classroom lessons, Community lessons, and Skills.
(Desktop view)



(Mobile view)

Third-Party Libraries

N/A“Service” Dependencies

N/A

Impact on Other Oppia Teams

As this is solely a redesign of the Learner dashboard, no new feature will be added or removed so there would be no impact on other Oppia teams.

Key High-Level and Architectural Decisions

Architecture decision depends on the mocks provided by Diana. Please see [here](#). Risks and mitigations

Potential Risk	Mitigation
User Confusion: One risk is that users may find the new design confusing or difficult to use, which could lead to a decrease in engagement or satisfaction.	To mitigate this risk, I'll request Diana to conduct user testing and understand the needs and preferences of users. Also, Júlia Falarini(designer) tests the design with the users on the weekend. (see here)
Compatibility Issues: The redesigned dashboard may not be compatible with all	To mitigate this risk, I'll test it across different platforms (like Chrome and Firefox) and devices(mobile, desktop) and ensure that the

viewports or browsers, which could limit user access and lead to frustration.	dashboard is optimized for a variety of user experiences.
---	---

Implementation Approach

Storage Model Layer Changes

The storage model layer will not be changed.

Domain Objects

The domain objects will not be changed.

User Flows (Controllers and Services)

The User Flows will not be changed.

Web frontend changes

The changes will be more clear by taking a look over these mocks ([here](#))

Prototype for this project: [see here](#)

HOW the feedback tab will be shifted to the profile-pic drop-down menu?

- **FeedbackUpdatesPage will be created in**
oppia/core/templates/pages/feedback-updates-page by creating the following file.
 1. Feedback-updates-page-root.component.html
 2. Feedback-updates-page-root.component.spec.ts
 3. Feedback-updates-page-root.component.ts
 4. Feedback-updates-page-routing.module.ts
 5. Feedback-updates-page.component.css
 6. Feedback-updates-page.component.html
 7. Feedback-updates-page.component.spec.ts
 8. Feedback-updates-page.component.ts
 9. feedback-updates-page.module.ts

- The drop-down menu is present in the `top-navigation-bar.component.html` and it contains all the links for the top navigation bar and for the drop-down menu present below the profile pic.
- Links present in `top-navigation-bar.component.html` have a similar code snippet, differences on “href” and smart-router

SEE:

```
<li>
href=" "
[smartrouterlink] = " "
</li>
```

- Constant will be initialized for the Oppia codebase in the “constants.ts”
Constants have a similar code snippet.

SEE:

```
"FEEDBACKUPDATES": {
  "ROUTE": " ",
  "TITLE": " ",
  "META": [
    {
      "PROPERTY_TYPE": "itemprop",
      "PROPERTY_VALUE": "description",
      "CONTENT": " "
    },
    {
      "PROPERTY_TYPE": "itemprop",
      "PROPERTY_VALUE": "og: description",
      "CONTENT": " "
    }
  ]
}
```

- Routing will be done in “app.routing.modules.ts”

```
{
  path:
```

```
AppConstants.PAGES_REGISTERED_WITH_FRONTEND.FEEDBACKUPDATES.ROUTE,
  pathMatch: 'full',
  loadChildren: () => import('feedback updates page module path')
    .then(m => m.Feedback Updates page module component name)
}
```

- The code for the feedback-updates-page will be utilized from the learner-dashboard-page.
- In the redesigned learner dashboard feedback updates is a page and when the user clicks on the feedback updates from the user profile pic drop-down menu the URL change to "localhost:8181/feedback-updates".
- Feedback-Updates-backend-api service will be created to get the data and accordingly changes will be made to the backend and **feedbackpageupdateshandler** will be used.
- Here are some steps to be taken
- Feedback_updates.py , feedback_updates_test.py file will be created.

```
FEEDBACK_UPDATES_DATA_URL = (
  '/feedbackupdateshandler/data')
```

- This URL will be created in feconf.py
- In the main.py

```
get_redirect_route(
  r'%s' % feconf.FEEDBACK_UPDATES_URL,
  feedback_updates.FeedbackUpdatesPage),
```

access decorator will be created in acl_decorator.py

- Below is the code snippet

```
def can_access_feedback_updates(
  handler: Callable[..., _GenericHandlerFunctionReturnType]
) -> Callable[..., _GenericHandlerFunctionReturnType]:
  """Decorator to check access to learner dashboard.
  .....
```

- In oppia/core/templates/domain/
- feedback_updates folder will be created it contains
- feedback-updates-backend-api.service and
- feedback-updates-backend-api.service.spec.ts

How do different components render according to the screen size?

Different components will be rendered according to screen size using the `*ngIf` directive.

`ngIf = "windowIsNarrow"` or `ngIf = "!windowIsNarrow"`

Initially this variable is initialized with false

```
windowIsNarrow: boolean = false;
```

“**WindowDimensionService**” is used to set the value of this variable.

```
this.windowIsNarrow =  
this.windowDimensionService.isWindowNarrow();
```

How the navigation bar will be created below the main navigation bar(for mobile view)?

The code present below is for the navigation bar present below the main navigation bar for the Home tab (in mobile view)

```
<div *ngIf="windowIsNarrow &&  
activeSection===LEARNER_DASHBOARD_SECTION_I18N_IDS.HOME">  
  </div>
```

Same code snippets for goal and progress tab

Use of “activeSection” and “activeSubsection”

→ “activeSection” is a variable that helps to switch between different sections.

There are three sections namely “Home tab”, “Goals tab” and “Progress tab”

In both mobile and desktop view.

Tab	Data needed	Endpoint URL
Home Tab	Continue where you left off→	

	Classroom Lessons→ <ul style="list-style-type: none"> • topicsToLearn • allTopics • partiallyLearntTopicsList • untrackedTopic 	<ul style="list-style-type: none"> • /learnerdashboardtopicsandstoriesprogresshandler/data
	Community lessons→ <ul style="list-style-type: none"> • IncompleteExplorationlist • IncompleteCollectionsList 	<ul style="list-style-type: none"> • /learnerdashboardexplorationsprogresshandler/data • /learnerdashboardcollectionsprogresshandler/data
	Learn some thing new section→	
	<ul style="list-style-type: none"> • untrackedTopics • We are getting the data of topic(Untracked topic) that are shown in Learn something new section and passed to the home-tab-component. 	<ul style="list-style-type: none"> • /learnerdashboardtopicsandstoriesprogresshandler/data
	Lesson you saved for later→ <ul style="list-style-type: none"> • explorationPlaylist • collectionPlaylist • passed to the home tab. 	<ul style="list-style-type: none"> • /learnerdashboardexplorationsprogresshandler/data • /learnerdashboardcollectionsprogresshandler/data
	Learner Groups→ <ul style="list-style-type: none"> • Learner group component will be used 	<ul style="list-style-type: none"> • /learner_dashboard_learner_groups_handler
Goal Tab	<ul style="list-style-type: none"> • allTopics • topicsToLearn • learntTopicsList • partiallyLearntTopicyclis • learntToPartiallyLearntTopics 	<ul style="list-style-type: none"> • /learnerdashboardtopicsandstoriesprogresshandler/data

	<p>Earlier we did not show the name of chapters but in new mocks, we are showing the chapters name and here is the data we are required to show them.</p> <pre>▼ (2) [StoryNode, StoryNode] ▶ 0: StoryNode {_id: 'node_1', _title: 'smalladd 1', _description: '', _desti ▶ 1: StoryNode {_id: 'node_2', _title: 'smalladd2', _description: '', _desti length: 2 [[Prototype]]: Array(0)</pre>	<ul style="list-style-type: none">• /learnerdashboardtopicsandstoriesprogresshandler/data
Progress Tab	<p>In Progress section→</p> <p>Classroom Lessons→</p> <ul style="list-style-type: none">• partiallyLearntTopic• topicsToLearn	<ul style="list-style-type: none">• learnerdashboardtopicsandstoriesprogresshandler/data
	<p>Community Lessons→</p> <ul style="list-style-type: none">• incompleteExplorationsList• incompletedCollectionsList	<ul style="list-style-type: none">• /learnerdashboardexplorationsprogresshandler/data• /learnerdashboardcollectionsprogreesshandler/data
	<p>Skills→</p> <ul style="list-style-type: none">• partiallyLearntTopicsList	<ul style="list-style-type: none">• /learnerdashboardtopicsandstoriesprogresshandler/data
	<p>Finished section→</p> <p>Classroom Lessons→</p> <ul style="list-style-type: none">• learntTopicsList	<ul style="list-style-type: none">• /learnerdashboardtopicsandstoriesprogresshandler/data
	<p>Community Lessons→</p> <ul style="list-style-type: none">• completeCollectionsList• completedCollectionsList	<ul style="list-style-type: none">• /learnerdashboardexplorationsprogresshandler/data• /learnerdashboardcollectionsprogreesshandler/data
	<p>Skills→</p> <ul style="list-style-type: none">• completedStoriesList• learntTopicsList	<ul style="list-style-type: none">• /learnerdashboardtopicsandstoriesprogresshandler/data

Feedback updates	This is already present in the learner dashboard.	<ul style="list-style-type: none"> • /learnerdashboardfeedbackupdateshandler/data
	<ul style="list-style-type: none"> • A New handler will be created and give the required data 	<ul style="list-style-type: none"> • /feedbackupdateshandler/data

Home Tab

★ Continue where you left off

```
<oppia-home-tab
[currentGoals]="topicsToLearn"
[goalTopics]="allTopics"
[partiallyLearntTopicsList]="partiallyLearntTopicsList"
[untrackedTopics]="untrackedTopics"
[username]="username"
(setActiveSection)="setActiveSection($event)">
</oppia-home-tab>
```

- This section will contain goal topics, Community lessons, and topics that are part of the learner group.
- **Goal topic:**
As from the above code `[goalTopics]="allTopics"`
All the topics in the Home tab are goal topics. So I will make the goal tag on the **Oppia-learner-story-summary-tile**.
- **Community lessons:**
- In the existing learner dashboard. We are getting all data of the community lessons in the learner-dashboard-page.component.ts file and then this data is passed to the community lessons tab.

```

<oppia-community-lessons-tab
[incompleteExplorationsList]="incompleteExplorationsList"
[incompleteCollectionsList]="incompleteCollectionsList"
[completedExplorationsList]="completedExplorationsList"
[completedCollectionsList]="completedCollectionsList"
[explorationPlaylist]="explorationPlaylist"
[collectionPlaylist]="collectionPlaylist"
[subscriptionsList]="subscriptionsList"

[completedToIncompleteCollections]="completedToIncompleteCollections">
</oppia-community-lessons-tab>

```

- Similarly, I will pass the [IncompleteExplorationlist] to the home tab.
And will show the **oppia-exploration-summary-tile** in the “Lessons in Progress section”.

By using node count, Completed node count, I get the number of chapters, Completed chapter respectively.

- And if the storyProgress variable I will differentiate tiles into two sections “In Progress” and “Recommended”

StoryProgress = 0. It goes to Recommended for You section. Button text (Start).

Story Progress !=0. It is in the Lessons in Progress section. Button text (Resume).

Please refer to the below code

→

```

for(var topicSummaryTile of this.continueWhereYouLeftOffList) {
for(var storySummary of topicSummaryTile.canonicalStorySummaryDicts) {
let stotyNodeCount = storySummary.getNodeTitles().length;
let storyCompletedNodeCount = storySummary.getCompletedNodeTitles().length;
let storyProgress = Math.floor(
(storyCompletedNodeCount / stotyNodeCount) * 100);

var storyData: storySummaryTile = {
  topicName: topicSummaryTile.name,
  storySummary: storySummary,
  storyProgress: storyProgress
}

if(storyProgress !==0) {

```

```

    this.storyInProgress.push(storyData);
  } else {
    this.storyInRecommended.push(storyData);
  }
}
}
}

```

I will use calculate the storyProgress by

```

this.nodeCount = storySummary.getNodeTitles().length;
this.completedNodeCount = storySummary.getCompletedNodeTitles().length;
this.storyProgress = Math.floor(
  (this.completedNodeCount / this.nodeCount) * 100);

```

By using node count, Completed node count, I get the number of chapters, Completed chapter respectively.

- And if the storyProgress variable I will differentiate tiles into two sections “In Progress” and “Recommended”

StoryProgress = 0. It goes to Recommended for You section. Button text (Start).
 Story Progress !=0. It is in the Lessons in Progress section. Button text (Resume).

★ Learn something new

- We will use the <search-bar> component (see “Reusable Components” table below).
- In the existing, we are using the **for loop with slice property** that shows the **three topics** and the 4th card link to the classroom. For the new redesign we will show all the topics and instead of 4th we will make a button.
- Topic from the classroom

This section is already present in the existing dashboard.

→ Lessons you saved for later

In the existing learner dashboard. We are getting all data of the community lessons in the learner-dashboard-page.component.ts file

```
this.learnerDashboardBackendApiService  
    .fetchLearnerDashboardCollectionsDataAsync()
```

And

```
this.learnerDashboardBackendApiService  
    .fetchLearnerDashboardExplorationsDataAsync();
```

- Is used to fetch the data for collection and Exploration.
and then this data is passed to the community lessons tab.

```
<oppia-community-lessons-tab  
[incompleteExplorationsList]="incompleteExplorationsList"  
[incompleteCollectionsList]="incompleteCollectionsList"  
[completedExplorationsList]="completedExplorationsList"  
[completedCollectionsList]="completedCollectionsList"  
[explorationPlaylist]="explorationPlaylist"  
[collectionPlaylist]="collectionPlaylist"  
[subscriptionsList]="subscriptionsList"  
[completedToIncompleteCollections]="completedToIncompleteCollections">  
</oppia-community-lessons-tab>
```

Similarly I will pass the

```
[explorationPlaylist]="explorationPlaylist"  
[collectionPlaylist]="collectionPlaylist"
```

- Data to the home tab component and use the **Oppia-exploration-summary-tile** and **oppia-collection-summary-tile** to show the cards (see “Reusable Components” table below for details).

Learner Group name on the tile

- The below step give the explanation to get the group title on the (**oppia-learner-topic-summary-tile**)

- By using the below steps learner group title will be stored corresponding to the story id in `Learner-dashboard-page.component.ts` and then this data is passed to home-tab and progress tab.
- I will fetch the learner group joined data from `/learner_dashboard_learner_groups_handler`
(`learnerGroupsJoined: ShortLearnerGroupSummary[] = [];`)

```

▼ (3) [ShortLearnerGroupSummary, ShortLearnerGroupSummary, ShortLearnerGroupSummary] ⓘ
  ▶ 0: ShortLearnerGroupSummary {_id: 'GzBmqXeWAHFj', _title: 'Group of facilator name JSRK', _des
  ▶ 1: ShortLearnerGroupSummary {_id: 'UYSZfKJttA0d', _title: 'Group of facilator name test1', _de
  ▶ 2: ShortLearnerGroupSummary {_id: 'WENTLhamIljE', _title: 'group 1 of shivv 1', _description:

```

- From above we get the id of all the learner joined id and this Id's will be stored in the array.
- The id of the learner groups will be passed to

```

this.learnerGroupBackendApiService.fetchLearnerGroupInfoAsync(
this.learnerGroupId
)

```

- It fetches the data for all learner groups from

```

/view_learner_group_info_handler/<learner_group_id>

```

And in response, I get

```

learnerGroupInfo(LearnerGroupData)

```

for all the groups and it contains title and story ids

- Then the story (story id) in the selected goal (topic) will compare with story ids in Learner group data stored.
- The titles (Learner group names) will be stored corresponding to the story id.

```

interface LearnerGroupStoryTitles {
  id: number;
  titles: string[];
}

```

Screenshot that shows the **Learner Group Data**.



```

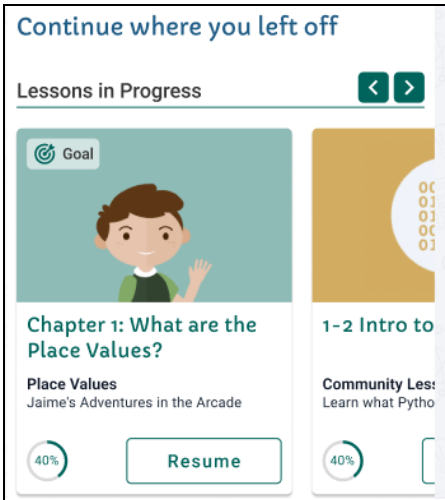
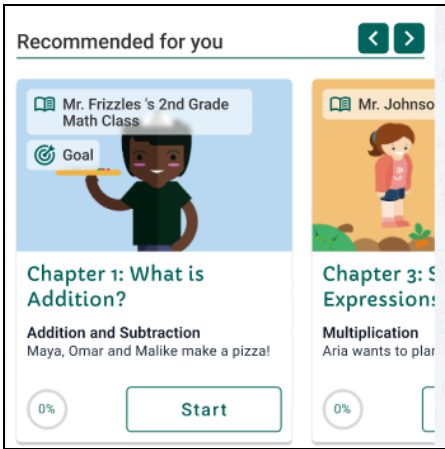
LearnerGroupData {_id: 'GzBmqXeWAHFj', _title: 'Group of facilator name JSRK',
  _learnerUsernames: Array(1), ...} ⓘ
  _description: "login with test@test.com\nI will invite learner(shivv)"
  _facilitatorUsernames: ['JSRK']
  _id: "GzBmqXeWAHFj"
  _invitedLearnerUsernames: []
  _learnerUsernames: ['shivv']
  _storyIds: ['gn9UrMKVPs7I']
  _subtopicPageIds: []
  _title: "Group of facilator name JSRK"

```

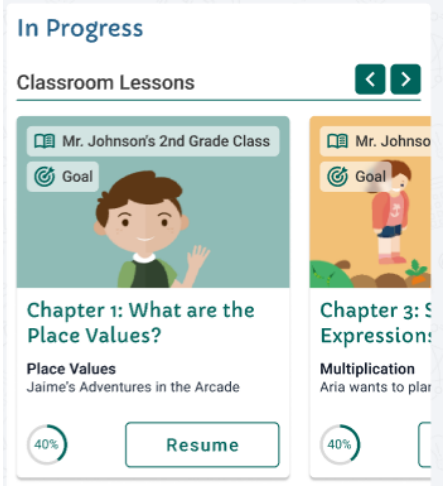
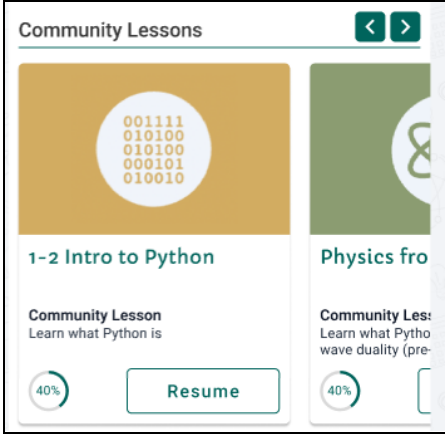
- A story can be in different groups. (i.e. a story can be included by different facilitators in their syllabus). In this case, the group title will change every second (i.e. fades in and fades out).

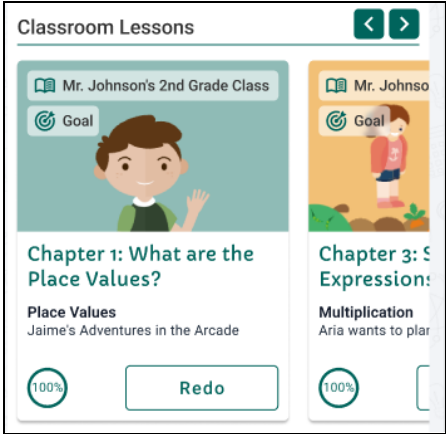
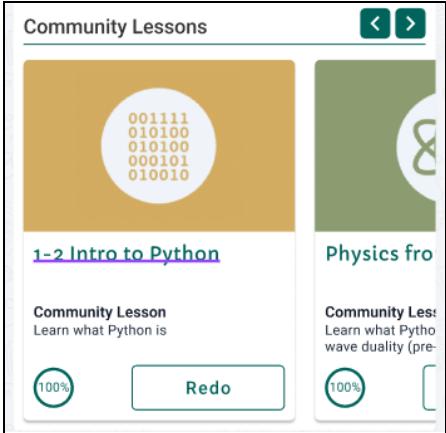
Reusable components:



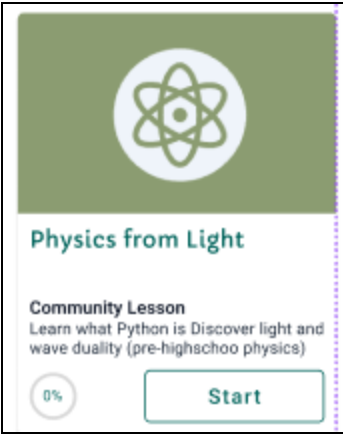
Component's tag name	New attributes	Usages	Customization
<oppia-search-bar>	Search-bar-color: The color of the search bar	Classroom page: 	search-bar-color-hex-code="#00FF00"
		Learner dashboard: 	search-bar-color-hex-code="#205D86"

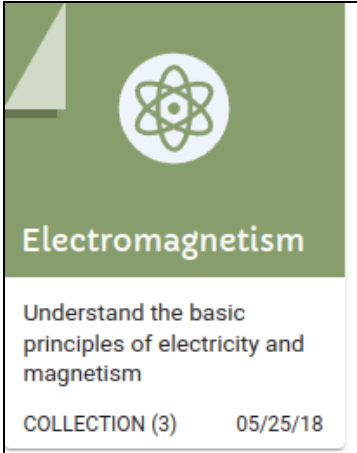
<p>[Carousel] <oppia-carousel-bar></p>	<p>ng-content</p>	<p>Learner dashboard page: Home tab → Continue where you left off</p> 	<p><oppia-carousel-bar> //for loop to show the Tiles of lessons in Progress <oppia-carousel-bar></p>
		<p>Learner dashboard page: Home tab → Recommended for you</p> 	<p><oppia-carousel-bar> //for loop to show the Tiles of recommended for you <oppia-carousel-bar></p>

		<p>Learner dashboard page: Home tab → Learn something New</p> <div><p>Topics available in Oppia's Classroom</p><div><div><div>500</div><div>6</div><div>2506</div><div>20000</div></div><div><div>Place Values</div><div>5 Lessons</div></div><div><div>Addition and Subtraction</div><div>7 Lessons</div></div></div><p>→ Or Explore All Math Lessons In Classroom</p></div>	<pre><oppia-carousel-bar> //for loop to show the Tiles of ultracked topic <oppia-carousel-bar></pre>
		<p>Learner dashboard page: Home tab → Lessons saved for play later</p> <div><p>Lessons you saved for later</p><div><div><div><div></div></div><div><div>Long Term Exercise Physiology</div><div>Community Lesson</div><div>Learn to explain how the human body adapts to regular exercise in the long...</div><div>0%</div><div>Start</div></div><div><div>1-2 Intro to</div><div>Community Les</div><div>Learn what Pytho</div><div>0%</div><div></div></div></div></div></div>	<pre><oppia-carousel-bar> //for loop to show the Tiles of lessons saved for you later <oppia-carousel-bar></pre>

		<div>Learner dashboard page: Progress tab → In Progress</div> <div></div>	<div><oppia-carousel-bar> //for loop to show the Tiles of Classroom lessons <oppia-carousel-bar></div>
		<div>Learner dashboard page: Progress tab → In Progress</div> <div></div>	<div><oppia-carousel-bar> //for loop to show the Tiles of community lessons <oppia-carousel-bar></div>

		<p>Learner dashboard page: Progress tab → Finished</p> 	<pre><oppia-carousel-bar> //for loop to show the Tiles of classroom lessons <oppia-carousel-bar></pre>
		<p>Learner dashboard page: Progress tab → Finished</p> 	<pre><oppia-carousel-bar> //for loop to show the Tiles of community lessons <oppia-carousel-bar></pre>
<pre><oppia-explorat ion-summary-til e></pre>	<pre>displayAreaIsLearn erDashboard: Whether the tile is being displayed in the context of</pre>	<p>Learner Dashboard page:</p>	<pre>displayAreaIsLearnerDa shboard = "true"</pre>

	the learner dashboard (Boolean)		
		Library page: 	displayAreaIsLearnerDashboard = "false".
<oppia-collection-summary-tile>	displayAreaIsLearnerDashboard: Whether the tile is being displayed in the context of the learner dashboard (Boolean)	Learner Dashboard Page: 	displayAreaIsLearnerDashboard = "true".
		Library Page:	displayAreaIsLearnerDashboard = "false".

		 <p>The image shows a carousel tile for 'Electromagnetism'. It features a green header with a white atom icon. Below the header, the text reads: 'Understand the basic principles of electricity and magnetism'. At the bottom, it says 'COLLECTION (3)' and '05/25/18'.</p>	shboard = "false".
--	--	--	--------------------

★ How to use the carousel at different places

Here ([see](#)) is the carousel prototype for the continue where you left off section.

Similarly will be done for others.

And the code snippet used to implement is →

IN the parent component→

```
<div class="continue-where-you-left-off-section"
*ngIf="continueWhereYouLeftOffList.length !== 0">
  <oppia-carousel-bar>
    <div *ngFor="let topicSummaryTile of continueWhereYouLeftOffList">
      <div *ngFor="let storySummaryTile of
topicSummaryTile.canonicalStorySummaryDicts; let i = index">
        <oppia-learner-story-summary-tile
[storySummary]="storySummaryTile" [displayArea]='homeTab'
[topicName]="topicSummaryTile.name">
        </oppia-learner-story-summary-tile>
      </div>
    </div>
  </oppia-carousel-bar>
</div>
```

In the child component→

```
<div class="oppia-coursel-container">

  <div class="left-right-button">
    <button mat-button class="md-no-ink oppia-carousel-scroller"
      (click)="scroll(true)">
      <i class="fas fal fa-angle-left"></i>
    </button>

    <button mat-button class="md-no-ink oppia-carousel-scroller"
      (click)="scroll(false)">
      <i class="fas fal fa-angle-right"></i>
    </button>
  </div>

  <div class="oppia-carousel-content">
    <div class="tiles">
      <ng-content></ng-content>
    </div>
  </div>

</div>
```

→ Learner Group

For the learner group subsection, the **oppia-learner-groups-tab component** will be used with the change in UI according to mocks

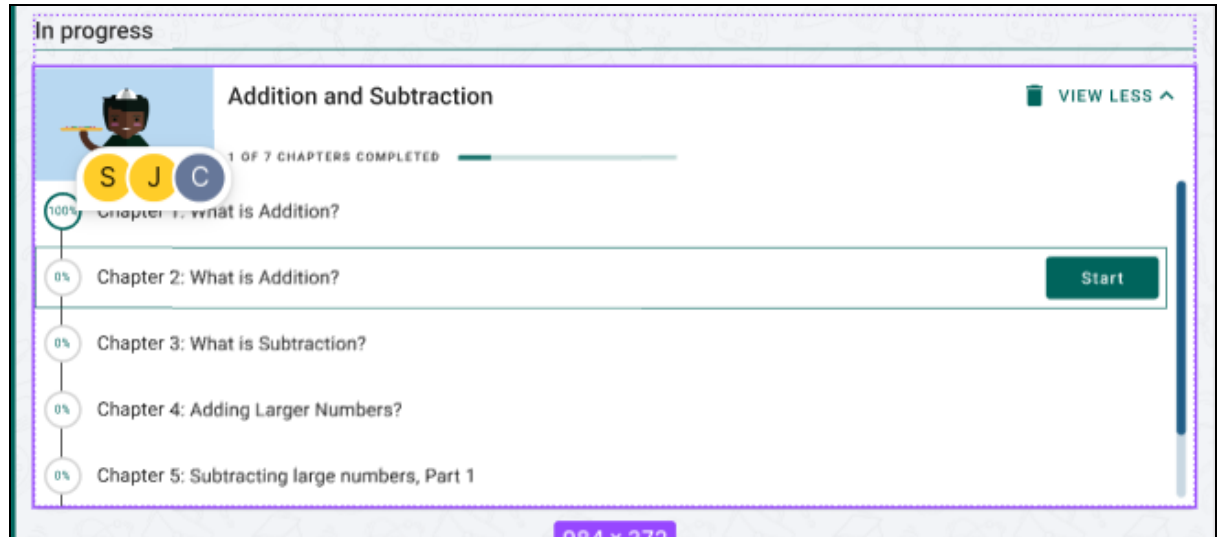
→

Goal Tab

★ **Current goal**

- “Add a goal” button will be implemented that opens a modal.
- A modal component (**add-or-edit-goal-modal.component**) will be created.

★ In Progress



- The new things from the current dashboard are:
- The new thing is the names of chapters, which I will get from Storynode



- Here title in the above image is the name of the chapters.

• For showing the progress of the chapter.

- For calculating the progress we need
 - 1) Total number of checkpoints
 - 2) Number of checkpoints learner has reached.

$$\text{progress} = \left(\frac{\text{Number of checkpoints learner has reached}}{\text{Total number of checkpoints}} \right) * 100;$$

- [Here](#) are the rough changes, that specify how did I get the data of checkpoints.
- In the specified link we have a function (`fetchProgressInExplorationsOrChapters`) in the (`this.readOnlyExplorationBackendApiService.`)
- `fetchProgressInExplorationsOrChapters` will take the array of the

- exploration Ids (expIds) and name of the user.
- And it redirects to the route

```
get_redirect_route(
    r'/user_progress_in_explorations_handler/<username>',
    learner_dashboard.LearnerExplorationProgressHandler),
```

- And this route map to the `LearnerExplorationProgressHandler` present in the `learner_dashboard.py`.
- `LearnerExplorationProgressHandler` will have one schemas for the exploration Ids.
- In this handler, I am getting the user progress in the explorations using the method `get_user_progress_in_exploration` that I have implemented in `exp_fetchers.py`.
- This function will take expIds (array of exploration Ids).
- How the function (`get_user_progress_in_exploration`) work.
 1. It generates combinations of the user ID with each exploration ID.
 2. Then It retrieves the exploration user data models for the user ID and exploration ID combinations.
 3. Then It initializes an empty list to store the progress summaries of all exploration. It loops through each user ID and exploration ID pair. For each pair, it retrieves the corresponding exploration object.

(Here, I map the user_id and exploration_id and store this mapped data in the list and then pass it to the `get_multi()`.
So that In the `get_multi()`, it can generate_id (of the form `[user_id].[exploration_id]`) and fetch the corresponding data from the `ExplorationUserDataModel`.)
 4. Then, It gets a list of all checkpoints in order for the exploration then It determines the count of visited checkpoints by finding the index of the most recently visited checkpoint in the list of all checkpoints.
 5. Then, It appends a dictionary containing the exploration ID, visited checkpoints count, and total checkpoints count to the progress list.
 6. After this all steps we return list of progress summaries.
- Import the `exp_fetchers` (from `core.domain import exp_fetchers`) In the `learner_dashboard.py` and pass the expIds and username to the `get_user_progress_in_exploration` present in the `exp_fetchers.py`. This will return the JSON data.

```
exploration_progress = (
```

```
exp_fetchers.get_user_progress_in_exploration(
    user_id, exp_ids))
```

- And then we will render this JSON data.

```
self.render_json(exploration_progress)
```

- And In the frontend we get data of checkpoints in the form of object in the same order as the exploration are pushed to the expIds array.

```
✖ ▶ ▼ [ChapterProgressSummary] ⓘ 'progressof exploration .....
```

→ For completed

As from the above image we have **completedNodeTitles**, so I will use this array for the completed section.



Progress


★ In Progress

In Progress

Classroom Lessons

Mr. Johnson's 2nd Grade Class

Goal




Chapter 1: What are the Place Values?

Place Values
Jaime's Adventures in the Arcade

40% [Resume](#)

Mr. Johnson's 2nd Grade Class

Goal




Chapter 3: Single Digit Expressions from 1-5

Multiplication
Ana wants to plant a garden

40% [Resume](#)

Mr. Johnson's 2nd Grade Class

Goal




Chapter 1: What is a Decimal?

Decimal
Reena's Race to the Finish

40% [Resume](#)

Mr. Johnson's 2nd Grade Class


Goal



Chapter 4: Finding a Percentage of a Number


Percentages
Monica Helps at the Jewelry Store

40% [Resume](#)

[Display More](#) 

[Hug x Hug](#)


Community Lessons



1-2 Intro to Python

Community Lesson
Learn what Python is


40% [Resume](#)



Physics from Light

Community Lesson
Learn what Python is Discover light and wave duality (pre-highschool physics)

40% [Resume](#)




Long Term Exercise Physiology

Community Lesson
Learn to explain how the human body adapts to regular exercise in the long...

40% [Resume](#)

Skills



The Place Names and Their Values

Place Values

40% [Resume](#)

- **Community Lessons:**
- In the existing learner dashboard. We are getting all data of the community lessons in the learner-dashboard-page.component.ts file and then this data is passed to the

community lessons tab.

```
<oppia-community-lessons-tab
[incompleteExplorationsList]="incompleteExplorationsList"
[incompleteCollectionsList]="incompleteCollectionsList"
[completedExplorationsList]="completedExplorationsList"
[completedCollectionsList]="completedCollectionsList"
[explorationPlaylist]="explorationPlaylist"
[collectionPlaylist]="collectionPlaylist"
[subscriptionsList]="subscriptionsList"

[completedToIncompleteCollections]="completedToIncompleteCollections">
</oppia-community-lessons-tab>
```

- **Classroom Lessons:**

- In the existing learner dashboard we are fetching the Topic and stories Data in the learner dashboard component.ts.

- |

```
et dashboardTopicAndStoriesDataPromise = (
    this.learnerDashboardBackendApiService
        .fetchLearnerDashboardTopicsAndStoriesDataAsync(),
```

- and we are getting the partially Learnt topic list

```
this.partiallyLearntTopicsList = (
    responseData.partiallyLearntTopicsList);
```

- So I will pass **partially LearntTopic** to **oppia-community-lessons-tab** for classroom lessons.

★ Skills

- We already using the skill in the Progress tab of the existing learner dashboard. I will use the existing.

★ Finished

- **Community lessons:**

In the existing learner dashboard. We are getting all data of the community lessons in the learner-dashboard-page.component.ts file and then this data is passed to the community lessons tab.

```
<oppia-community-lessons-tab
[incompleteExplorationsList]="incompleteExplorationsList"
[incompleteCollectionsList]="incompleteCollectionsList"
[completedExplorationsList]="completedExplorationsList"
[completedCollectionsList]="completedCollectionsList"
[explorationPlaylist]="explorationPlaylist"
[collectionPlaylist]="collectionPlaylist"
[subscriptionsList]="subscriptionsList"

[completedToIncompleteCollections]="completedToIncompleteCollections">
</oppia-community-lessons-tab>
```

- **Classroom Lessons:**

- In the existing learner dashboard we are fetching the Topic and stories Data in the learner dashboard component.ts.

```
let dashboardTopicAndStoriesDataPromise = (
  this.learnerDashboardBackendApiService
    .fetchLearnerDashboardTopicsAndStoriesDataAsync(),
```

- and we are getting the partially Learnt topic list

```
this.LearntTopicsList = (
  responseData.LearntTopicsList);
```

So I will pass **LearntTopic** to **oppia-community-lessons-tab** for classroom lessons.

Skills:

We already have the skills. It is not new.

Documentation changes

There will be no documentation changes

Testing Plan

E2e testing plan

#	Test name	Initial setup step	Step	Expectation
1.	Check where you left off test, Learn something new, "Edit goal", and "Current goal" section test	Login (by default you are on the learner dashboard, and the Home tab will be open) (Initially the number of the topic in the "Suggested for you section" is zero and the number of the completed story is also zero) Create a topic with the title "Topic 1".	Navigate to "Topic and Skill Dashboard"	There is 1 topic there. Its name is "Topic 1".
			Navigate to the Home section	The topic "Topic 1" is in the "Suggested for you" section.
			Navigate to the Goal section Click on the Add goal button.	A modal will open. It should have a topic "Topic 1" in the "Edit Goal" section.
			Add "Topic 1 " as a current goal and navigate to the goal section.	It should have a topic "Topic 1" in the "In Progress " section of the goal tab.
			Navigate to the Home section.	It should have a topic "Topic 1" in the "Continue where you left off" section.
3.	Lessons	Login as "creator 1" and	Click on logout.	"Creator 1" should get logout

	saved for later	create two explorations (with the titles "fraction","singing", and categories mathematics", and "music" respectively	Login as "Learner 1"	"Learner1 " will get login.
			Open the Community Library. Mark the lessons (fraction and singing)to play later.	The "Fraction" and "Singing" lessons will appear in the "Lesson to save later section" of the Home tab.
			Open the Home Tab of the learner dashboard	The explorations "fraction" and "singing" will be shown in the "Lessons saved for later section" of the Home tab.
			"Learner 1" click on logout.	"Learner 1" should get logout.
5.	Progress tab test, In Progress, Skills, Finished.	Log in with the email progresstabtest@example.com .	Navigate to Creator Dashboard.	There should be an exploration (BUS101) in the library page.
			Create an exploration with the title "BUS101".	
			Click on Logout.	Creator (Progresstabtest@example.com) should be logout.
			Log in with the email "learner@progresstab.com"	"learner@progresstab.com" Should get login.
			Navigate to the community library page. Search for exploration (BUS101)	There should be an exploration (BUS101) in the library page
			Click on the lesson. Then start and complete one card.	One card of the exploration should get completed.
			Navigate to the Progress tab of the learner dashboard.	There should be an exploration (BUS101) in the In Progress section of the Progress tab.
			Navigate to "Topic and	There is 1 topic there. Its

			Skill Dashboard"	name is "Topic 1".
			Navigate to the progress tab	The topic name "Topic 1" is in the skill section of the progress tab.
			Complete the topic and Navigate to the Progress tab.	The topic name "Topic 1" is in the completed section of the progress tab.
			Navigate to the Goal tab	The topic name "Topic 1" is in the completed section of the Goal tab.
			Click on logout	Learner(learner@progresstab.com) will be logout.
7	Feedback Update test	After login, Create an exploration (name of exploration BUS101).	Click on the " Create exploration " button after moving to Creator Dashboard and create and publish an exploration.	It should create and publish the exploration named BUS101.
		After login, Then move to the feedback update page and check the feedback	Click on exploration, and type feedback. (i.e Complete an exploration and give feedback.)	It should play and complete the exploration with feedback.
			Click on feedback updates from the profile pic drop-down menu Click on the feedback thread.	It should open the feedback updates page and match the feedback messages.

Manual testing for Learner Group

#	Test name	Initial setup step	Step	Expectation
1.	Learner Group test	Login as "creator 1"	Navigate to the "Topic and Skills Dashboard"	There is 1 topic there. Its name is "Topic 1".
			Click on logout	"Creator 1" should logout.

		Login as "Facilator 1" (i.e login with email (learnergroup@facilator.com))	Navigate to the "Facilator Dashboard"	"Facilator 1" should navigate to the "Facilator Dashboard"
			Click on the "Create Group" button	It should naviage to "create learner group" page
			Add the Group Details	It should add the Group title as "LearnerGroupTitle" and Group Description as "LearnerGroupDescription"
			Click on "Next" button	It should navigate to the "Add Syallabus Items"
			Search the topic ("Topic 1")	It should show the topic ("Topic 1")
			Click on the "Add to Syllabus" button	It should add the "Topic 1" in the learner Group syllabus.
			Click on "Next" button	It should navigate to "Invite learners"
			Search the learner name (i.e learner 1)	It should find the learner ("learner 1").
			Click on the "Invite Learners" button	It should invite the learner to the group.
			Click on the learner "Create Group" button	It should create a Group.
			Click on logout	"Facilator 1" should get logout.
		Login as "learner 1" (i.e login with email) (learnergroup@learner.com) (By default you are on home tab)	Click on the "Accept" button	A modal should open with the info regarding group and having "Join Group" button.
			Click on "Join Group" button	A Group should be added to the "Your Group section"

Implementation Plan

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

Milestone 1

Key objective for this milestone: The feedback tab will be removed from the learner dashboard and added to the profile pic dropdown menu. An indicator will be added to notify the user when there are updates. The architecture for the home tab will get redesigned ([according to mocks](#)). The project will ensure that the feedback flow is fully functional end-to-end. All flows will work correctly on desktop and mobile.

No.	Description of PR / action	Prereq PR numbers	Target date for PR creation	Target date for PR to be merged
1.1	Learners will be able to use Feedback Updates from the profile-pic drop-down menu (Feedback Updates will be moved out of the learner dashboard menu and added to the profile pic drop-down menu.) With a feature gating system to gate the enabling of the new shifted feedback update with a feature flag (name of the feature flag will be show_feedback_updates_in_profile_dropdown). (Note: The end-to-end test for this PR will be written with the PR itself)	-	8 June	15 June
1.2	Implementing the feature gating system to gate the enabling of the new redesigned learner dashboard with a feature flag (name of the feature flag will be show_redesigned_learner_dashboard)	1.1	18 June	22 June
1.3	Revamping the underlying architecture of the Learner Dashboard menu and "Continue where you left off" section and update the Architecture of "Learn	1.1,1.2,	29 June	4 July

	something new" section			
	PM demo Date: 6 July			
1.4	Introducing the Learner group in the Home tab (Note: end-to-end test for 1.3,1.4 will be written within this PR.)	1.1,1.2,1.3	8 July	14 July

Milestone 2

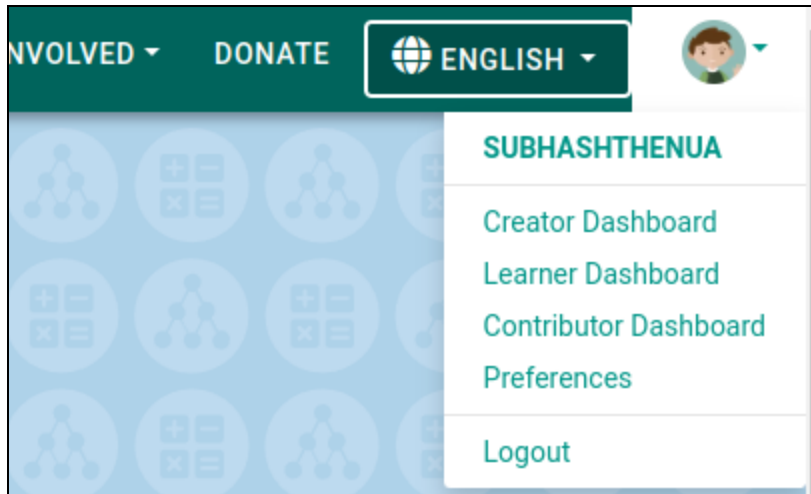
Key objective for this milestone: The Goals and Progress tabs will get redesigned as per the mocks ([here](#)) to provide learners with a new and consistent dashboard that will improve their learning experience. All flows will work correctly on desktop and mobile.

No.	Description of PR / action	Prereq PR numbers	Target date for PR creation	Target date for PR to be merged
2.1	Update the architecture of the Goals Tab to match the new mocks.	-	5 August	12 August
2.2	Update the underlying architecture of the Progress Tab to match the new mocks. (Note: end-to-end test for 2.1,2.2 will be written within this PR.)	2.1	4 Sept	14 Sept
	PM demo Date: 15 Sept			

The Goal Tab and Progress Tab cannot be split into multiple PRs because all of its subsections are interconnected and depend on each other. Therefore, the entire tabs need to be revamped as a whole.

★ **Technical plan to create a new learner dashboard, page (Instead of using many if/else conditions)**

→ A new learner dashboard page will be created and it will be accessed from



→ **NewLearnerDashboard** component will be created

1. new-learner-dashboard.component.css
2. new-learner-dashboard.component.html
3. new-learner-dashboard.component.spec.ts
4. new-learner-dashboard.component.ts

- There will be no change in the backend file of the learner dashboard.
- All the handlers and APIs will remain the same.
- Only a new component will be created and a feature flag (**show_redesigned_learner_dashboard**) will be used to hide the new page.
- I will make two main div's in learner-dashboard.component.html
In the first div, I will put all the code of the learner dashboard.
and in the second div, I will render the new-learner-dashboard component
and control the divs using the feature flag
- In the **learner-dashboard-page.component.html**

```
<div *ngIf="!isShowRedesignedLearnerDashboardFeatureFlagEnabled()">
  //code of existing learner dashboard.
</div>
<div *ngIf="isShowRedesignedLearnerDashboardFeatureFlagEnabled()">
  // new learner dashboard component
```

```
<oppia-new-learner-dashboard></oppia-new-learner-dashboard>

</div>
```

→ In the **learner-dashboard-page.component.ts**

```
isShowRedesignedLearnerDashboardFeatureFlagEnable(): boolean {
    return
    this.platformFeatureService.status.ShowRedesignedLearnerDashboard.isEnabled
;
}
```

→ Above point help to render the new page on the same URL (/learner-dashbaord).

→ Then we follow the further steps of the feature gating system (Described below).

★ Details for using a feature flag

- FOR Milestone 1
- The **Feedback_Updates_Page** feature flag will be used in the Production stage, as it can be activated in any of the development, testing, or production environments
- The Shifted feedback updates pull request (PR) will be merged with the new feature flag to the codebase, ensuring that the new feedback updates are gated behind the feature flag and not visible to users until it is fully shifted and ready for production
- The final pull request (PR) will be merged that finish the shifting of feedback updates, including changes that move the feature flag to the test stage. This ensures that the new feedback updates will be available in the testing environment and tested before making it available to users in the production environment.

- I will request the QA team to test the shifted feedback updates on the test server. If any issues are identified during testing, I will address them and fix them. Once all issues are resolved and the feedback update is ready for production, a pull request (PR) will be merged to “un-gate” the feature flag and move it to the deprecated stage, as well as remove any remaining references to the feature flag from the codebase.
- Once the feedback updates have been moved to the profile pic drop-down menu, **we can launch and make it available to users.**

- **The below feature flag will be added in the first milestone after moving the feedback update to the profile pic drop-down menu. This feature flag will then be used to facilitate the work of redesigning the home tab and completing the entire second milestone.**
- The **new_learning_dashboard** feature flag will be enabled in the production stage, as it can be activated in any of the development, testing, or production environments.
- The initial pull request (PR) will be merged to add the new feature flag to the codebase. After that, all PRs that redesign the learner dashboard will be merged, ensuring that the new learning dashboard is gated behind the feature flag and not visible to users until it is fully redesigned and ready for production.
- The final pull request (PR) will be merged that finish the redesign of the learner dashboard, including changes that move the **feature flag to the test stage**. This ensures that the new learner dashboard will be available in the testing environment and tested before making it available to users in the production environment.
- I will request the QA team to test the redesigned learner dashboard on the test server. If any issues are identified during testing, I will address them and fix them. Once all issues are resolved and the learner dashboard is ready for production, a pull request (PR) will be merged to “un-gate” the feature flag and move it to the deprecated stage, as well as remove any remaining references to the feature flag from the codebase.

- The Home Tab, Goal Tab, and Progress Tab cannot be split into multiple feature flags because all of its subsections are interconnected and depend on each other. Therefore, the entire tabs need to be revamped as a whole.

Future Work

In the future, it is possible to perform a redesign of other dashboards as well.

Note : I have taken references from [Krishita Jain 's proposal](#) while preparing the proposal.
(Thanks to “ krishita Jain”).