

Oppia Project Proposal

Personal Information:

Name: Avijit Gupta

Email: 526avijit@gmail.com

GitHub handle: 526avijitgupta

Project

Project Name: Creator dashboard

Project Description:

Oppia's tagline says - "Intelligent Online Learning". Learners can learn interactively through lessons created by the lesson creators, called explorations. This GSoC project aims to build a dashboard for the lesson creators to be involved in learning through student feedback and hence be able to act upon the same. This would make the learning process to be more transparent and the flow of information to be bidirectional. The quality of educational content will be much higher rate after each iteration. The analysis of statistical data will help lesson creators detect and respond to common misconceptions, so that students who encounter similar issues in the future can receive better feedback.

Structural Layout and Integration of the Dashboard:

On the **design part**:

- The design mockups for the dashboard are currently being developed. So it makes sense to follow the [latest](#) ones as a reference.

- I would like to propose the preliminary **design mockups for mobile**, classified on the basis of screen sizes. (Please Note that these are just the very basic preliminary designs which will be improved upon with the help of community feedback. Also, it's the first milestone to finalise these by the time the coding period begins) :



Mobile Design Mockups for Creator Dashboard

On the **back-end**:

- Controllers (for handling requests) for the dashboard and the relevant tests will be written (dashboard.py and dashboard_test.py under oppia/core/controllers/)
- The business logic and the core functionality will be handled by the combination of domain objects being acted upon by services. (creator_dashboard_domain.py and creator_dashboard_services.py under oppia/core/domain/).

- Both the domain objects and the services will be accompanied by their tests.
(creator_dashboard_domain_test.py and creator_dashboard_services_test.py under oppia/core/domain/)
- There appears no need to create a new model for the data. The existing models(statistics model) will suffice in providing the data for "Average Rating", "Total no. of plays", "Feedback", "Last Updated", "Completion Rates" and "Unhandled Answers".
- The Common misconceptions will be shown in the form of "Unhandled Answers" for which some additional events will be required to categorize them by state, since currently the number is the total number of unhandled explorations across all states.
- The creators will be provided with an option to manage an Unhandled Answer 2 ways - either ignore the answer altogether or to request feedback from the learner regarding the same.
- A job will be added to keep track of the data relative to the previous week and also to refresh it weekly.

On the **front-end**:

- The relevant page(s), controllers and karma tests will be written inside creator_dashboard/ directory under core/templates/dev/head/
- The dashboard will be accessible on `/creator_dashboard` or any other similar route.



Other than that, **end-to-end tests** will be written/modified accordingly.

Working of the dashboard:

The data required to be displayed in the dashboard is already available, but it needs to be aggregated. Currently, this data is recorded for each exploration separately. As a part of implementation of the project, a job will be created on the backend which will aggregate the data for all explorations of the creator for the ongoing week. It'll keep the data of the previous week

intact, so that a simple difference can determine the +/- for the fields “Average Rating” and “Total Plays”. As specified earlier, there will be no need to create a new model using this approach.

Clicking on the **Unhandled Answers**, the creator will be taken to a new page. This page will contain the name of the exploration and the details of the unhandled answers, (categorized by state) in a list view. The list view will contain the attributes: “Unhandled Answer Text”, “No. of times”, “Name of the State”, and a cross button(to remove the unresolved answer). It will be a modified version of the currently displayed format:

Unhandled Answers		
This2	1 time	
this3	1 time	
Yayy!		

The cross button will simply ignore the answer, but there will also be a functionality to enable the creators to **address the unhandled answers**. An “address” button will be displayed against each row for which the “number of times” is at least 2. Clicking on the address button will trigger a notification to the learner(s) who have given that answer, asking them for feedback regarding the same. This will be done through a new event that is triggered on clicking the “address” button.

The creator will be **notified only when there is new feedback or there are new unhandled answers**, not otherwise.

As the desktop [designs](#) show, there will be a “Sort By” option on the page, which will allow the creator to be able to **prioritize the data** on the basis of all the such attributes: Rating, Plays, Unhandled Answers, Feedback and Last Updated.

What interests you about this project:

I am interested in this project since it's a full stack project which aims to provide a platform for the lesson creators to improve the quality of content based on actual statistics. Not only does this project involve writing code on frontend and backend, but also provides an opportunity to participate in the design process.

Also, it amazes me how the creation of a dashboard without the change in any of the data models can have such a great potential impact on the quality of content available.

What areas of research will you do in this project:

- From what all domains will the data be aggregated.
- To what extent the current statistical data is queryable.
- Figure out what additional events can help record data (after discussion with community members).

Outline of a project plan and implementation schedule

- Community bonding period.
 - Get familiar with the code base, specifically the statistics part.
 - Discuss the laid out mobile designs and improve them by the process of iteration.
 - Achieve the above by working with the mentor(s) and other community people interested in the project.

Milestone 1: Finalize the desktop and mobile designs before the coding period begins.

- 3-4 weeks: Work on backend
 - Start by creating a job that aggregates the data and refreshes it weekly (also recording the relative progress for "Average Rating" and "Total Plays").
 - Create the domain, services and controllers required for the dashboard.

- Handle event(s) to notify the learner requesting feedback regarding an unhandled answer.
- Write the corresponding server-side tests for the same.

Milestone 2: Complete the additions to the backend and make the data ready to be served.

- 3-4 weeks: Frontend work
 - Create the frontend (including the mobile implementation) according to the finalized designs.
 - Write the corresponding client-side tests for the same.

Milestone 3: Working prototype; Ready for e2e testing.

- 1 week: End-to-end testing.
- 1 week: Ensure that the dashboard is complete, take community feedback for any further changes.
- 1 week: Documentation.

The above plan could go as expected or invariably re-distribute among the tasks, but a positive margin of time left would be great for starting Phase 2 work.

Related Technical Skills

I am currently pursuing my Bachelor's in Computer Science Engineering from IIIT, Noida, India. I'm mostly interested in web and mobile development using JavaScript and Python. I love doing front-end development, scripts that make life easy and working with moving parts while developing.

I've got experience in both Python and AngularJS, the former can be seen on my [github](#) repositories list, and the latter has been gained through a couple of internships that I have done, primarily working on AngularJS and its derivatives (Angular Material, Ionic Framework). Also, my [stackoverflow](#) profile showcases by interest and the required level of proficiency in the same.

Open Source and Education

I am one of the coordinators of the **Open Source Developers Club** in my university . Everything I develop and use is open source. As a contributor,

- I have contributed a few [patches](#) and reported a few [bugs](#) to Oppia project.
- I have contributed to a few [patches](#) to the Mozilla Firefox (gecko) project.
- I have contributed to the MozDef (Mozilla Defense) Project with a few [features](#) and also contributed to a few patches to their existing codebase under Mozilla Winter of Security 2015.
- All my other projects can be found on my [GitHub](#) profile.

I've also attended a few open source meetups including PyCon India 2015, and local meetups about JavaScript, Python etc.

The very philosophy of open source fascinates me and contributing to it is my passion, primarily because it feels nice to surrounded by a bunch of like-minded people! Also, I find it to be helpful in developing a great learning experience from the best of the developers in the world for free.

Summer Plans

Timezone: UTC+5.5(IST - India)

Typical working hours: 11 am to 1 am until July, 5 pm to 2 am from August.

Communication

Location: India

Preferred Communication Method: Gitter

Communication Frequency: twice a week (flexible)