

Migrate Problem Creator to Vue.js + Typescript

ARITRA

aritraramajumder8438@gmail.com

<https://github.com/aritra8438>

omegaUp account: [aritraramajumder8438](#)

Technical skills

I am currently a computer science sophomore pursuing a Master's degree in Science at Chennai Mathematical Institute. I completed my Bachelor's degree in Engineering from Jadavpur University.

The previous year, I participated in GSoC, '23 with Mathesar and it was a wonderful experience learning from them. The project was successful and can be accessed [here](#) (and a [short extension](#) for known bugs introduced by the PRs). I used to maintain bi-weekly blogs for my progress which you can find [here](#). I have also participated in HacktoberFest, '23.

I am an open-source developer and contributed to many repositories. I have listed below my experiences and skills. Besides, I'm also (at least used to be) a competitive programmer who loves Maths :3 I'm an [expert](#) on Codeforces.

Time	Organization	Role	Skills
2023, May.	Mathesar (GSoC, '23)	Student Developer	Python, Django, TypeScript, SvelteJS, PostgreSQL
2022, June	PricewaterhouseCoopers	Data Analytics	Python, Azure Data Factory
2023, September	Meshery	Developer Intern	Go, JavaScript
2023, March	Shiksha Sopan - IIT Kanpur	Backend Developer Intern	Python, Django.
2022-2023	Numerous projects	NA	JavaScript, VueJS, Python, Django

You can find out more about me on [LinkedIn](#), [Github](#) and [my portfolio](#).

Link to omegaUp merged PRs:

1. [Modify the problem edit page.](#)
2. [Modify the contest edit page.](#)
3. [Render initial tab based on anchor.](#)
4. [Modify error message in case of invalid parameter exception.](#)

This might get updated, so [here](#) you can find the updated list of merged prs.

Education

- Chennai Mathematical Institute
- Computer Science
- August 2023
- May 2025

Motivation

One of the main components of the **OmegaUp** platform is the competitive programming online judge. An online judge has 3 main components: problem, code and the verdict of the code on that problem. Currently, to create a problem on OmegaUp, we have to rely on the zip file that can be created through the expertise of the problem creators (which makes it difficult for the new users) or via an [external website](#). The [Github repo](#) that deployed the site is public and now a **Public Archive**. So, it can't be an official tool for OmegaUp. Moreover,

- If a user finds a bug on the website, as it's a public archive, it's not possible to make a pull request.
- No newer feature can be added to the website.
- The fate of the site largely depends on the author's availability.

Currently, after a problem is created, if we need to modify the same, we need to again upload the zip file to the CDP, modify it and upload the modified zip file on the problem-edit page on OmegaUp.

The screenshot shows the 'Edit problem' page for problem 'abcd' on the OmegaUp website. The page has a blue header with the OmegaUp logo and navigation links: Contests, Courses, Problems, Ranking, and Help. On the right of the header are a notification bell, a user profile icon labeled 'abc', and a power icon. Below the header, the page title is 'Edit problem abcd – Go to problem', with a 'Go to problem' link. A 'Help' link is also present. A row of buttons includes 'Edit problem' (highlighted in blue), 'Edit statement', 'View version history', 'Edit Solution', 'Admins', 'Tags', 'Download', and 'Delete'. The main content area is titled 'First time creating a problem? Here is how to write the file'. It contains two sections: 'Basic information' and 'Validation'. The 'Basic information' section has fields for 'Title' (containing 'abcd'), 'Alias' (containing 'abc'), 'Problem source' (containing 'abcd'), and 'Attach zip file' (with a 'Choose File' button and 'No file chosen' text). The 'Validation' section has a 'Problem type' dropdown menu (showing 'C, C++, C#, Java, Kotlin, Python, Ruby, Pascal, Haskell, L') and an 'Output validator type' dropdown menu (showing 'Token by Token').

Enabling users to directly generate a problem (without generating the intermediate zip file) may help a lot of users and make the edit problem action much easier and intuitive.

So, the motivation behind this project is migrating the whole CDP to the OmegaUp website. This project aims to make the OmegaUp problem creation easier, more intuitive and more versatile as well as backwards-compatible.

Detailed project description

Currently, we have two ways to create a problem for OmegaUp:

- Using [OmegaUp CDP](#), where we can write problems, solution code and test cases, and then the CDP generates a **Zip file**.
- We can manually create a **Zip file** for a problem (requires a lot of expertise).

Subsequently, [here](#) we can upload the **Zip file** and fill in the necessary info to create a problem.

The project aims to implement an official CDP by migrating the unofficial CDP to OmegaUp. The migration was already started by Mau and the implemented page can be viewed [here](#).

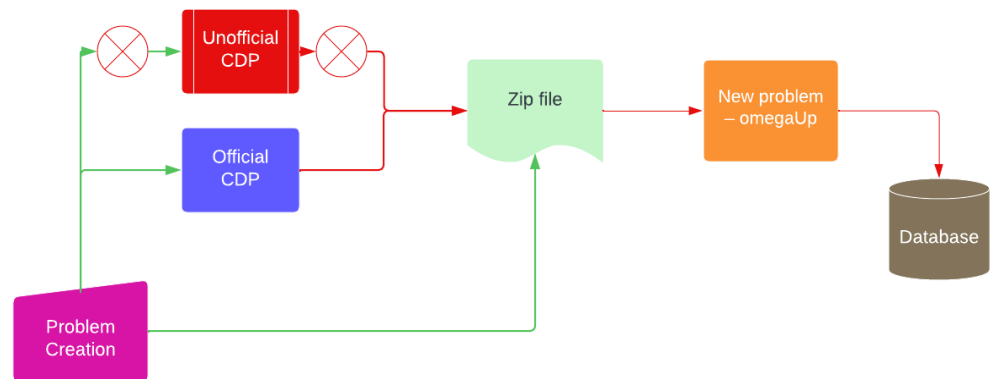
Go to File -> Make a copy and start writing your proposal

The screenshot shows the omegaUp problem creator interface. At the top, there's a navigation bar with the omegaUp logo and links for Contests, Courses, Problems, Ranking, and Help. Below this, there's a header with the user's name 'aritraramajunder8438' and a power button. The main area has a 'Name' field set to 'New Problem' and three buttons: 'Load Problem', 'Generate Problem', and 'New Problem'. Below the header, there are tabs for 'Statement', 'Code', 'Test cases', and 'Solution'. The 'Test cases' tab is active. On the left, there's a 'Groups' section with an 'Add' button. The main content area shows the 'Add' modal for creating a new case. It has three tabs: 'Case', 'Group', and 'Multiple cases'. The 'Case' tab is selected. It contains a 'Case Name' field with a note 'Lowercase and without whitespaces', a 'Group Name' dropdown menu set to 'No group', and a checkbox for 'Automatically calculate points (recommended)' which is checked. Below the checkbox, there's a small text explaining the formula: 'The program will calculate the points of this case with the next formula: (group points - (sum of cases inside the group that have explicitly defined points)) / (number of cases inside the group that do not have explicitly defined points)'. At the bottom of the modal are 'Cancel' and 'Add' buttons.

I would like to divide the project into two halves:

1. For the first half of the project, I would like to complete the migration of the CDP which will mimic the unofficial CDP and then generate a **Zip file**. Then the user will follow the same process to upload the file along with filling in the other info to generate a problem.

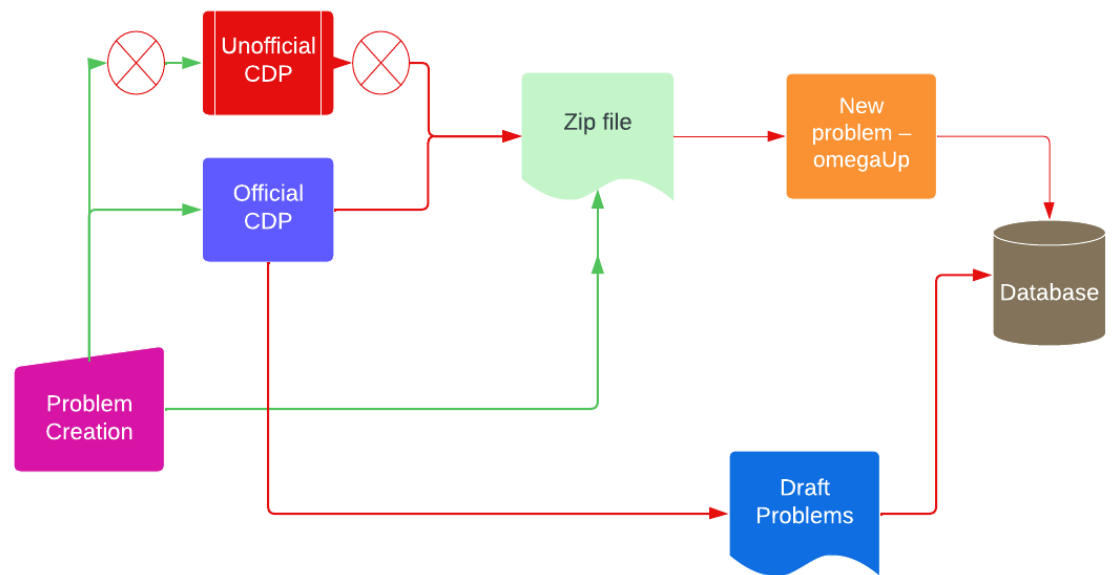
For this half, we can follow the structure of the unofficial CDP, as most of the users are used to that. The unofficial CDP is written in TypeScript and React, Redux is used to implement the store and tests are written in Cypress. To migrate, we have to use the VueJS framework that will replace React and Vuex will replace Redux.



2. For the second half, I would like to integrate the CDP directly with the problem generation (i.e. optionally remove the dependence on the intermediate process that creates a zip file). As some users are accustomed to problem generation by creating a zip file, we would keep the process backwards-compatible.

Currently, there is no persistence in the CDP page (i.e. no way to store an incomplete problem, we have to download the zip and again upload to continue uploading). The new functionality should be able to store draft problems (incomplete problems) and create problems directly, as we want to make the problem creation part independent of the intermediate zip file. **If we plan on removing the intermediate zip file, the reason behind saving draft problems becomes necessary, as if a user wants to create a problem in steps, he /she might need to store the incomplete problem as a draft and again start modifying it from where he/she has left off. (Currently, the same can be done by downloading the zip file for the incomplete problem and then again uploading it to modify.)**

Go to **File** -> **Make a copy** and start writing your proposal



Progress and my plans:

[Here](#) is the list of PRs Mau initiated as part of the migration. The components for the **Test Cases** tab are implemented. Mau also added tests for the same.

Subsequently, we would like to add the remaining tabs in the order **problem statement**, **solution code** and **editorial** and add tests for them. Once the CDP can completely handle writing a problem, we can start working on the **Generate problem** button.

Once the CDP is completely migrated, we will move to phase 2.

Frontend

The first part (i.e. the migration) needs no to very little changes in the server and no changes in the database.

For the second part,

```
CREATE TABLE `Problems` (  
    `problem_id` int NOT NULL AUTO_INCREMENT,  
    `acl_id` int NOT NULL,  
    `visibility` int NOT NULL DEFAULT '1' COMMENT '-1  
banned, 0 private, 1 public, 2 recommended',  
    ...  
)
```

We would like to modify the problems table so that it can store both the completed and draft problems. They can be distinguished by modifying the **visibility** column, which might then take **3 draft**. The reason behind the **draft option** is that we are allowing users to save incomplete problems as draft problems.

```
CREATE TABLE `Problems` (  
    `problem_id` int NOT NULL AUTO_INCREMENT,  
    `acl_id` int NOT NULL,  
    `visibility` int NOT NULL DEFAULT '1' COMMENT '-1  
banned, 0 private, 1 public, 2 recommended, 3 draft',  
    ...  
)
```

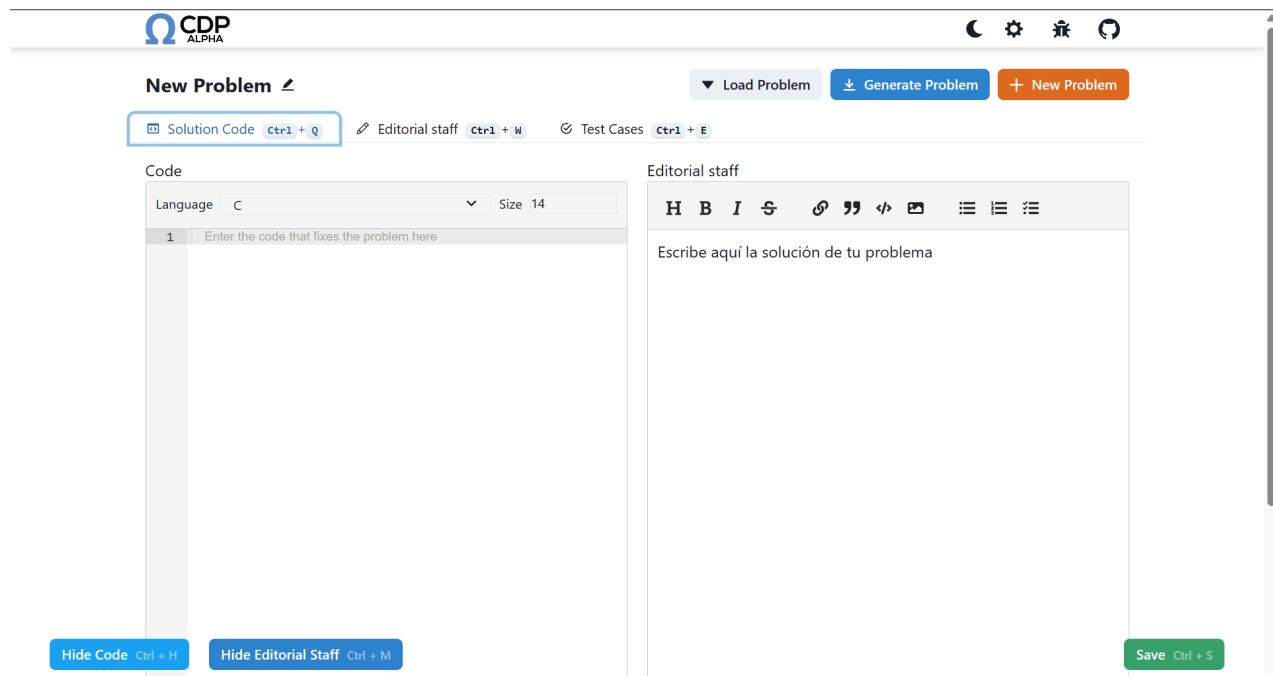
Subsequently, there will be a similar change in **abstract class Problems** [here](#). This may lead to a number of changes to the code executed on the server.

User experience

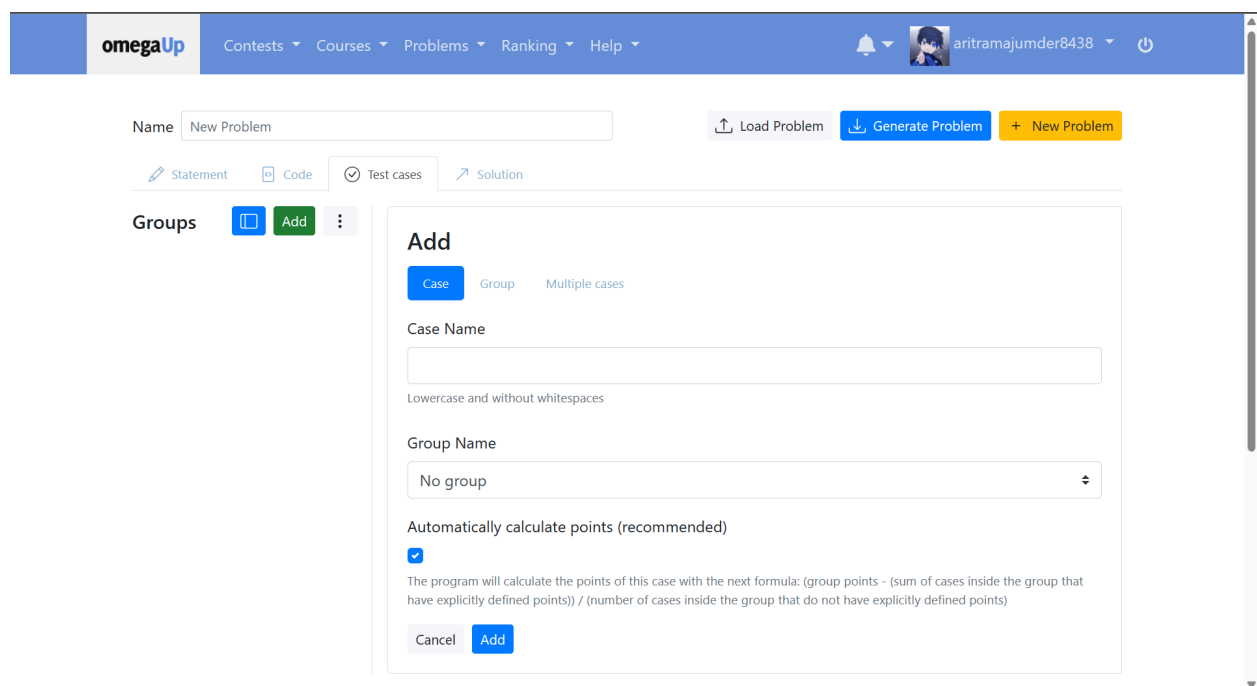
Let's look at the unofficial CDP and the current (incomplete) CDP:

- Unofficial CDP:

Go to File -> Make a copy and start writing your proposal



- Official CDP:



After the successful completion of the project, the official CDP should work like the unofficial CDP (i.e. create a zip file by clicking generate problem). Currently, on the official CDP page, only the **Test Cases** tab is implemented, the remaining task will be to complete the other 3 tabs (**statement**, **code**, **solution**) and add functions to the buttons. At the end of phase 1, we

Go to File -> Make a copy and start writing your proposal

will also include a button on the **problem/new/** page that will point to the newly created official CDP.

The screenshot shows the 'Create problem' form in the omegaUp interface. The form is titled 'Create problem' and includes a link for first-time users: 'First time creating a problem? [Here is how to write the file](#)'. The form is divided into sections: 'Basic information' and 'Tags'. The 'Basic information' section contains four input fields: 'Title', 'Alias', 'Problem source', and 'Attach zip file'. The 'Attach zip file' section has a 'Choose File' button and a 'No file chosen' text. A blue box with an arrow points to the 'Choose File' button, with the text 'To CDP' written next to it. The 'Tags' section has a message 'It's required to select at least one public tag' and a 'Public tags' input field.

Once the CDP is completely migrated to the official CDP page, the implementation of the second part will need a few UI changes. One tentative change might be the introduction of the **Save** and the **Save draft** button on the problem creator (**CDP**) page. Both the buttons need other necessary info (like, problem name, alias, source, tags etc.) to be filled out first on some modal.

Save button: Saving a problem will require validating the problem and then saving the problem to the database.

Save draft button: This doesn't require validating the problem. The sole purpose of this button is to save incomplete problems so that users don't have to start over and they can continue editing the problem from where they left off. Currently, we need to save the zip file of the incomplete problem from the CDP and upload it again to modify it.

Go to File -> Make a copy and start writing your proposal

omegaUp

Contests Courses Problems Ranking Help

aritraramajunder8438

Name New Problem

Load Problem Generate Problem + New Problem

Statement Code Test cases Solution

Groups Add

Add

Case Group Multiple cases

Case Name

Lowercase and without whitespaces

Group Name

No group

Automatically calculate points (recommended)

☒

The program will calculate the points of this case with the next formula: (group points - (sum of cases inside the group that have explicitly defined points)) / (number of cases inside the group that do not have explicitly defined points)

Cancel Add

Save

draft

Another tentative UI change might be the introduction of the process to directly edit a problem from the **edit problem** page so that users can continue editing without requiring a zip file.

Alternatives considered

Previously, I considered migrating the official CDP by getting rid of the zip file creation. In that case, problem creation might be temporarily unstable and **NOT** intuitive to use (Many users may not be accustomed to new functionalities). But, as already mentioned, we want to keep the process backwards-compatible, so we will keep the provision of creating problems through zip files even when OmegaUp has the new functionality of direct problem creation and editing.

There were a few alternatives in my mind (like using Redux etc.) but almost all of them are not as suitable for omegaUp as the current proposal is.

Security impact

This project has no impact on security. After the completion of the project, there will be some client-side changes, a few database and server side changes which merely incorporate new security issues.

Deployment plan

Normal deployment

Testing plan

Like other components, the problem creator component will have 'cypress' e2e tests and '@vue/test-utils' test.

Schedule

May 2	Proposal accepted or rejected.	<ul style="list-style-type: none">• Community Bonding Period - Discussing the project with mentors and finalizing the project implementation.• Familiarize myself with the codebase and Mau's official CDP.
May 27	Pre-work complete	Coding Officially begins!
Jun 6	Milestone #1	<ul style="list-style-type: none">• Implement the statement tab and write tests for it
June 13	Milestone #2	<ul style="list-style-type: none">• Implement the code tab and write tests for it.
June 20	Milestone #3	<ul style="list-style-type: none">• Implement the solution tab and write tests for it.
June 27	Milestone #4	<ul style="list-style-type: none">• Make all the buttons of the official CDP page functioning.• Write a blog to cover all the modifications we have done so far to the CDP page.
July 8 - July 12	Phase 1 Evaluation	<ul style="list-style-type: none">• During this period, I need to submit a report to my mentors on the performance of the official CDP page.• Start working on the process of removal of the Zip file.
July 19 July 26	Milestone #5	<ul style="list-style-type: none">• Modify the database and abstract class for problems to include the provision for draft problems.• This may need a lot of subsequent changes.

August 2	Milestone #6	<ul style="list-style-type: none">• Modify the problem creator UI to add save and save draft buttons.
August 9	Milestone #7	<ul style="list-style-type: none">• Modify the problem edit page to allow users to modify problems without creating a zip file.• Write a second blog to cover all the features we added to the problem creator page.
August 19 - August 26	Milestone #8	<ul style="list-style-type: none">• During this period, I need to submit the final report to my mentors on the performance of the official CDP page.• Write documentations on how to create a problem using new functionalities.
September 3 - November 11	Milestone #7	<ul style="list-style-type: none">• As this is a long project, we might need to extend the project beyond the initial coding phase.• In this period, all the uncompleted and blocked works will be finished.• Need to extensively check if the new functionality is working correctly.

This timeline is tentative and adjustable based on the mentors' feedback, advice and current progress.