Visual Redesign – For better programming

I. The Problem

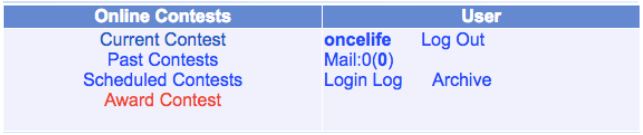
"Before anything else, preparation is the key to success". As a CS major student, computer algorithms play a crucial role in almost every aspect of life: from coursework to interview and job requirements. If one wants to become a successful programmer, mastering this field is a necessity, and online programming contests provides an easy and efficient way of practicing. From a survey made by freeCodeCamp, these are the top 10 coding contest website in 2017 (https://medium.freecodecamp.org/the-10-most-popular-coding-challenge-websites-of-2016-fb8a5672d22f). But there are far more good websites beyond this list. Today I will look into one of those, the Peking University Online Judge, analyzing its existing design shortages, and do a comprehensive responsive redesign.

As a platform dated back to 2003, Peking University Online Judge (Url: http://poj.org) has been a famous coding challenge website for a long time. Now it owns over 700k+ registered users, and thousands of daily submissions. However, there are some significant shortages on its visual design that affects its learnability and efficiency. Besides, it somehow lacks responsiveness and visual aesthetics.

II. Usability Analyze

Starting from a visual designer's point of view, let's first look into the website's usability shortage using the generally-accepted usability criteria:

- **Intuitive Design**
  The webpages contain large amount of small font texts, but don't have apparent layout structure; so, it is not easy for user to quick understand the structure hierarchy.

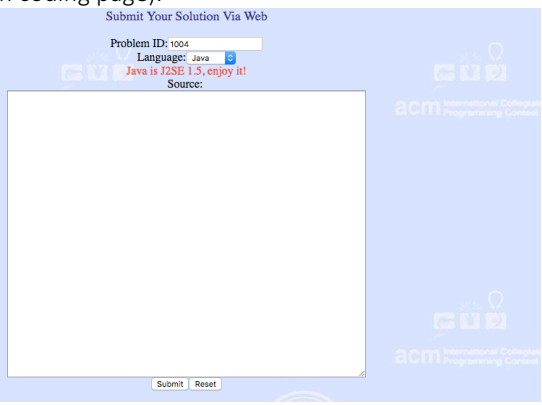


Everything is listed out. Using dropdown and only remain key categories gives better hierarchy

- **Ease of Learning**
  Not easy to learn for new users, because there's no highlight on core part of website (e.g. coding problems), and its layout is different to other prominent online coding websites.

All problems are in a large list, no categories or basic introduction (although many problems' names don't state themselves)

- **Efficiency of Use**
  It lacks some functionality that affect even experienced users (e.g. category of problems, problem description on coding page).



The area on the right, which supposed to be a problem description on the coding page for reference, is a whole blank block for now

- **Memorability**
  Most information in lists and table, hard to retrieve and memory the content.

- **Error Frequency and Severity**
  Most texts are small and no enough space in lists and tables, so it's easier for user to click a wrong position

POJ Challenge Test Round 1(This contest is hosting in poj.openjudge.cn)
ACM Trainning
IOI Training
ACM Training 2012.7.27
ACM Training 2012.7.26
ACM Training 2012.7.25

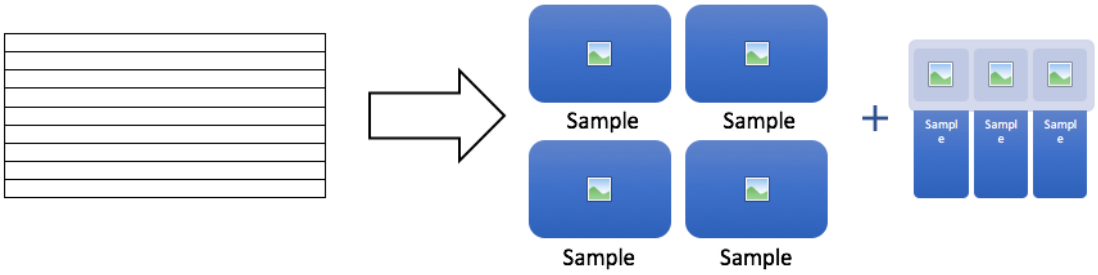
There are totally no space between adjacent topics

- **Subjective Satisfaction**
  This website lacks efficiency of use, and has a high error possibility; with that the satisfaction will be strongly affected.

III. Design considerations

When considering the design components, my primary goal is to fix the following problems of the original coding page:

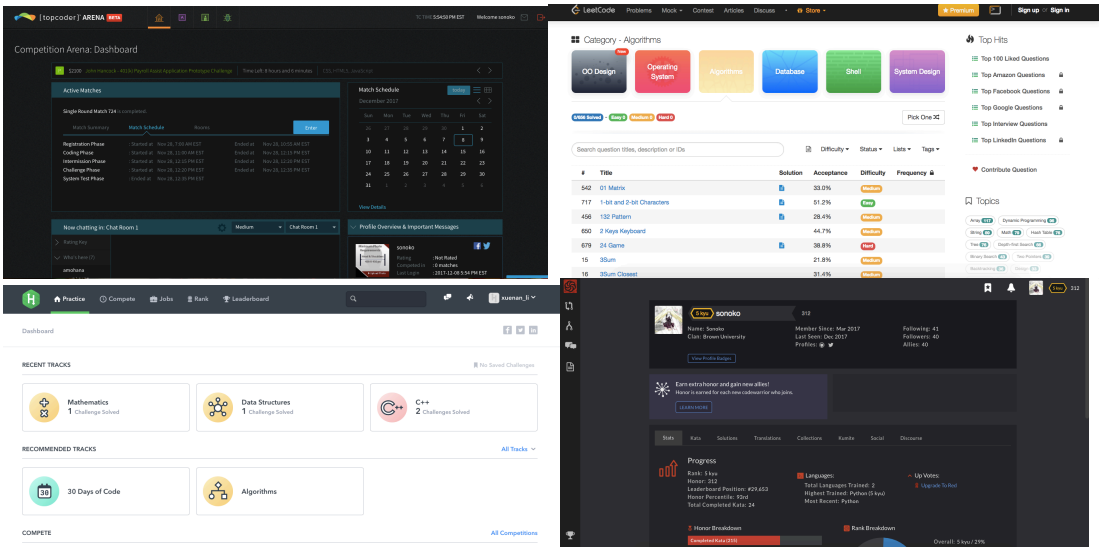
1) **Lack of structure and information**
   On original pages, almost everything is in plain tables. By changing these tables into dropdowns and panels and adding some necessary statistics or description, I achieve two goals: highlighting the purpose of each area, and allowing user to learn faster.



2) Unlikeness to its counterparts

As a website serving on very specific purposes, we can confidentially assume that the users own background knowledge on its functionality, and are likely to have usage habit got from using its counterparts. Thus, I was trying to changing its style to matching common representations in online coding websites. Example includes the overall minimalist style, displaying statistical info, or using light color for description and dark one for actual coding area.

Screenshot of POJ Homepage

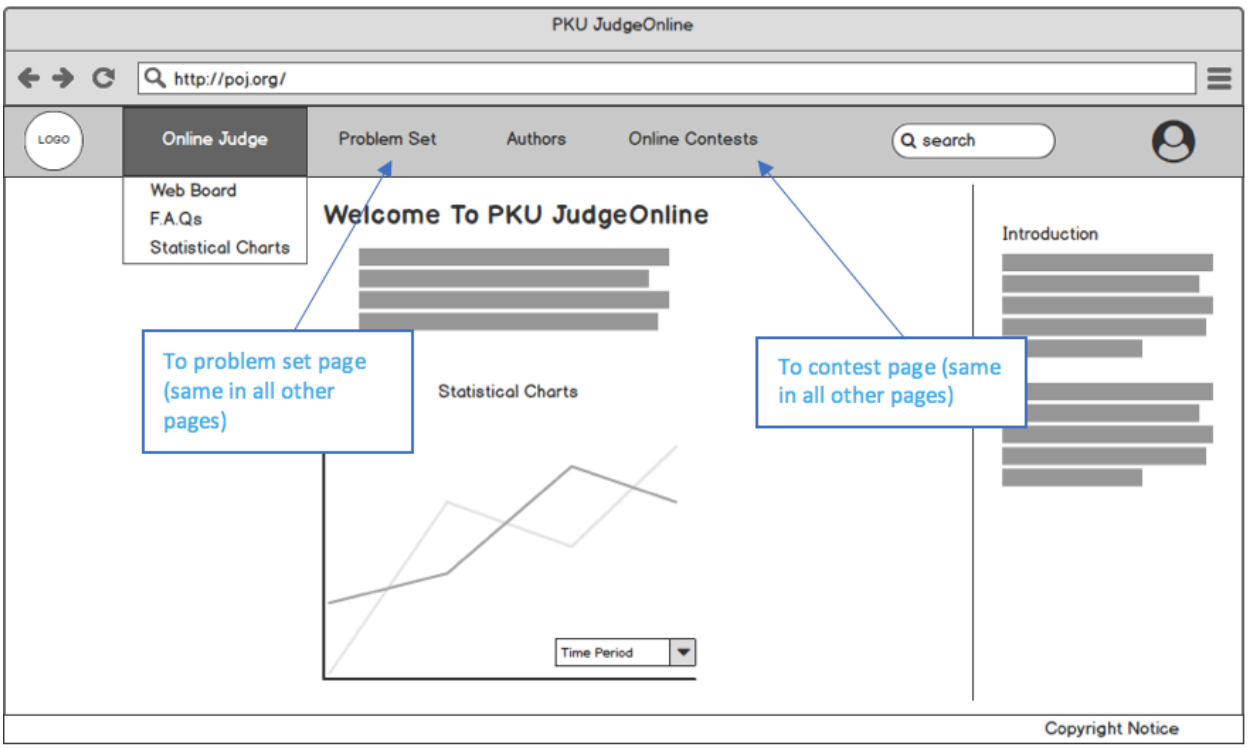


Screenshot of other famous coding websites

IV. Wireframing

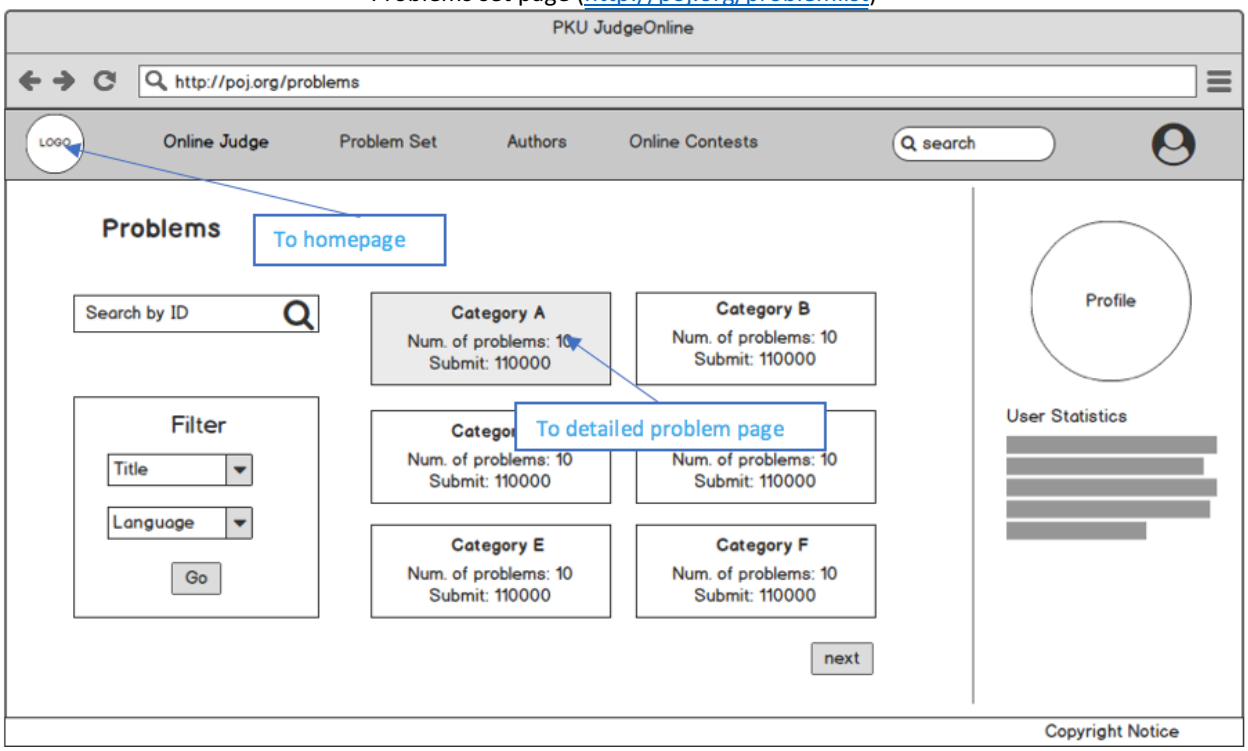
First step of the redesign process is creating wireframes. I redesigned 5 main screens, which are: homepage, problems set page, detailed problem page, coding page, and contest page, and adding comment with arrows to show the navigations between pages. (Design Tool: Balsamiq)

Homepage (http://poj.org)



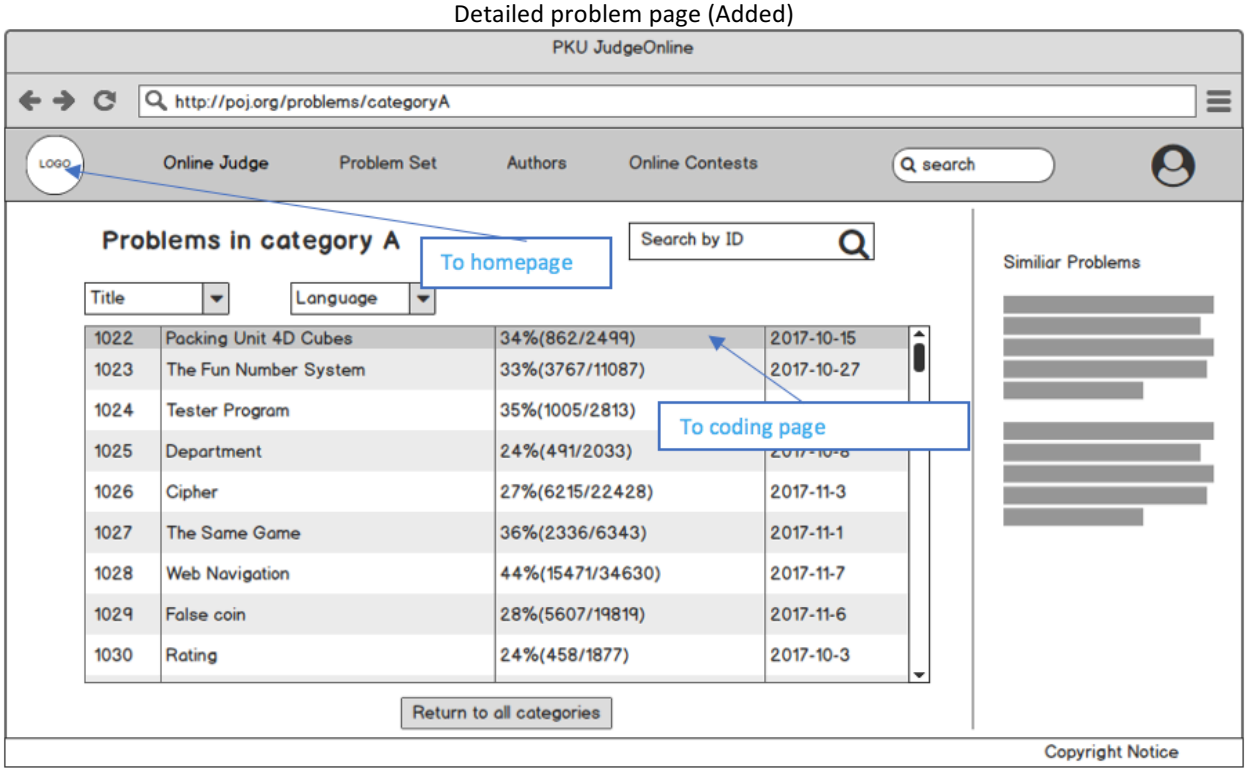
On original homepage, the navigation bar is a table of every options. I change this into dropdowns. I move the statistical chart from one of the subpage to the homepage. Generally, I design in purpose of making it clearer.

Problems set page (http://poj.org/problemlist)



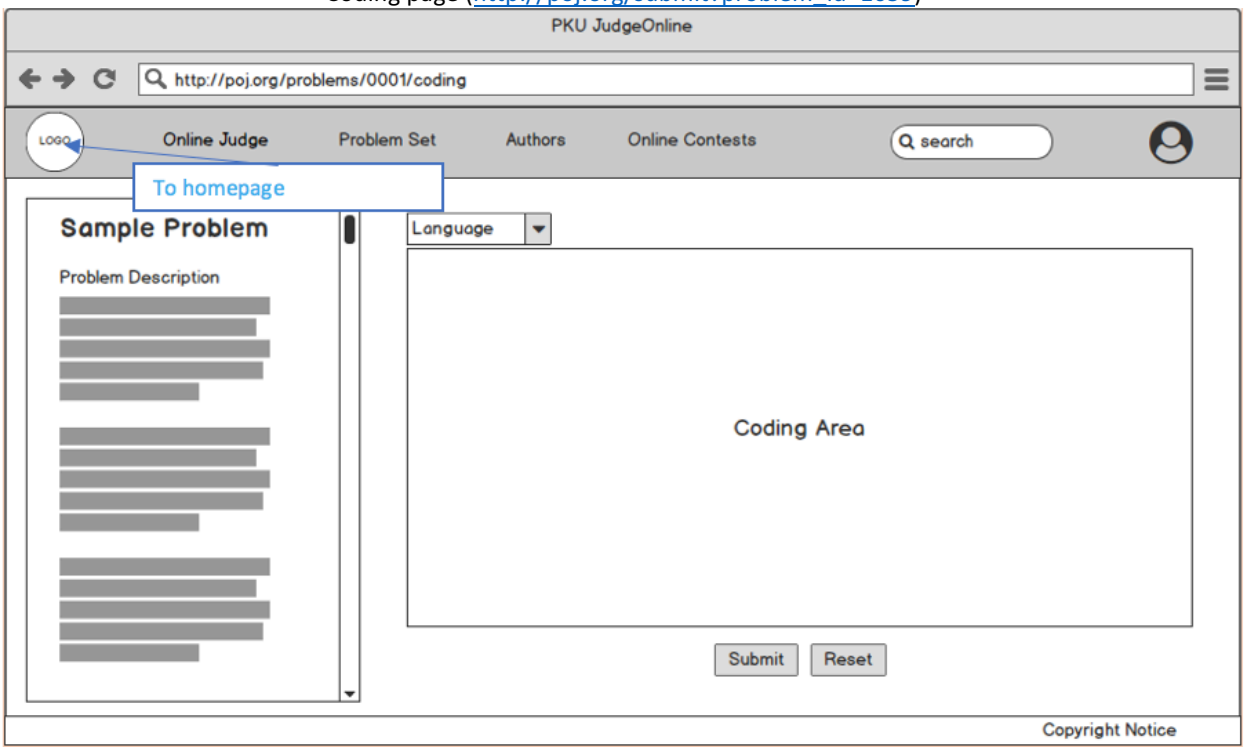
Original page lists all problems in a long list. I modify it into clickable pads based on category. I also add a user profile part for user to see his/her progress so far.

Detailed problem page (Added)



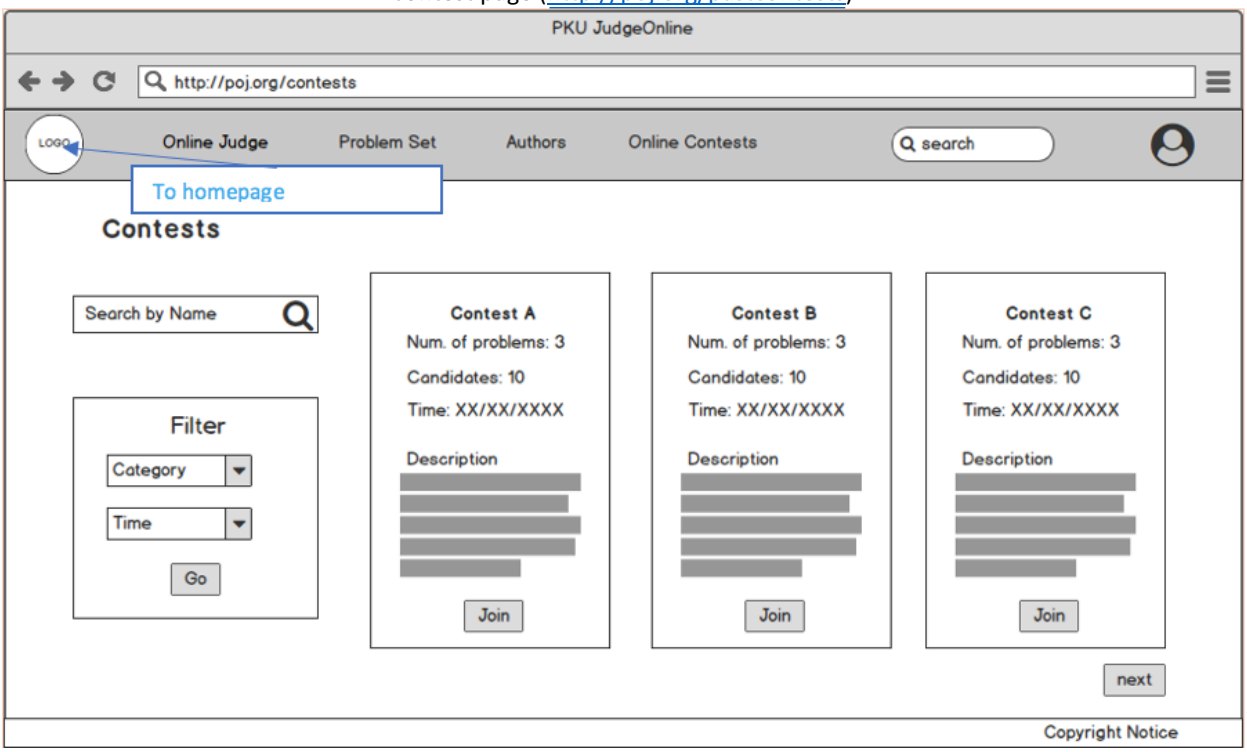
This page is added by me. It lists all problems in a certain category. And I add a suggestion part to provide similar questions to user based on their current choice.

Coding page (http://poj.org/submit?problem_id=1059)



Original coding page doesn't have problem description for reference (problem description is on another separate page). I combine them together.

Contest page (http://poj.org/pastcontests)



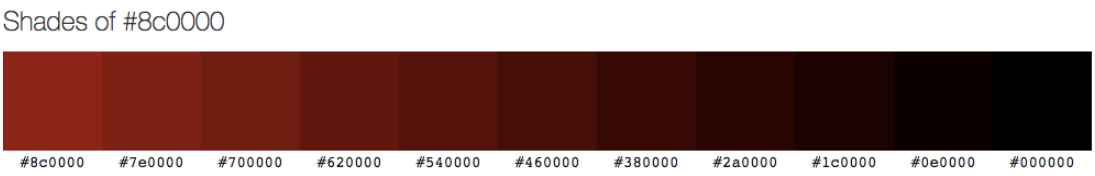
The original contest page list all contests in a single list. I change them into info pads, therefore there is space for basic description for each contest, which makes user easier to choose which to join.

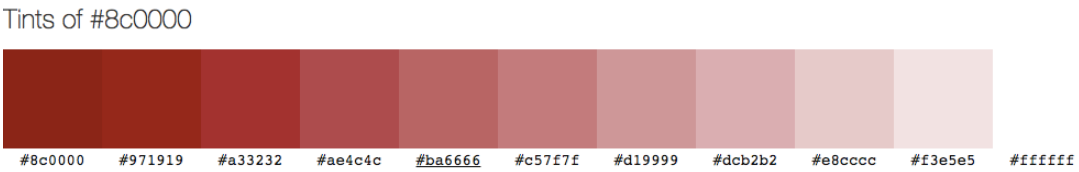
V. Hi-fidelity Mockup

Choose a Color Palette

I use color Red #8C0000 as primary color of this whole website, as it is the official color of Peking University. For the palettes, I use a website called Color Hex (http://www.color-hex.com) for help, by input the main color onto it. The other colors I choose (e.g. #666666, #f3e5e5 or white and black) are based on main color's shades/tints as well as Color Hex suggestions.

Shades and tints of main color





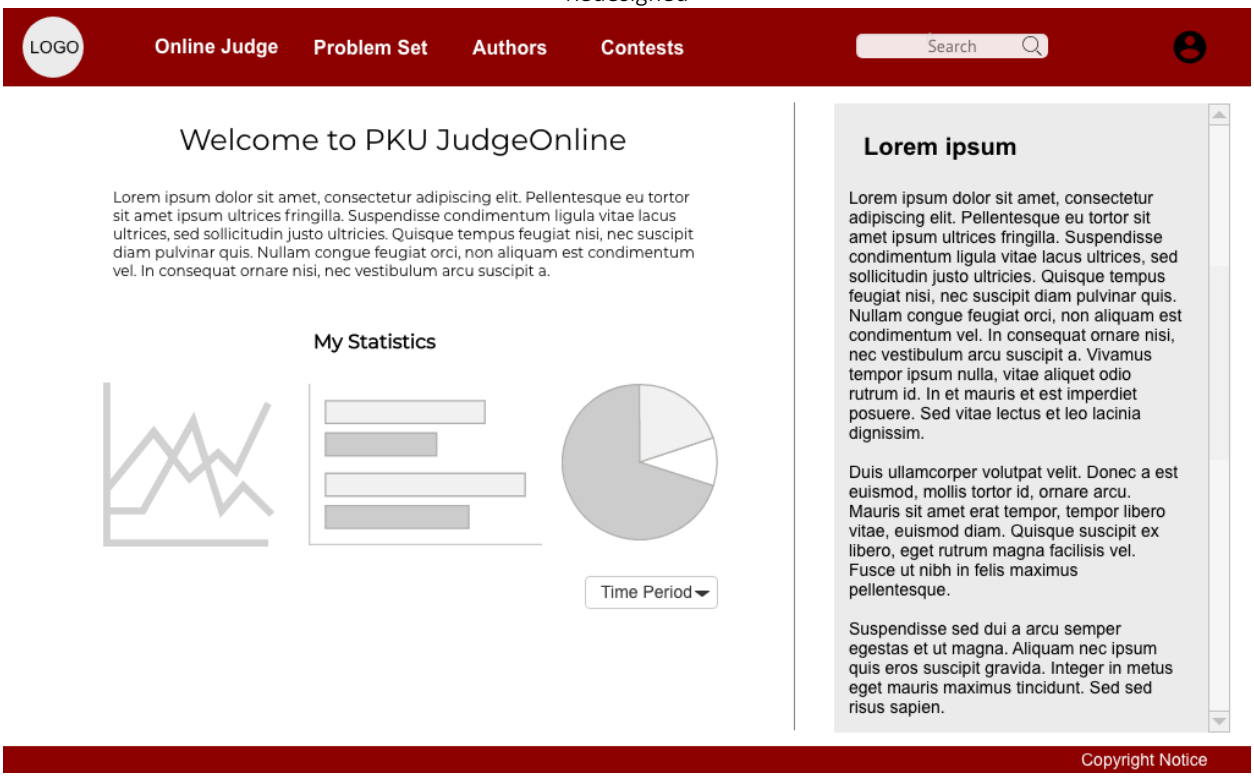
Mockups

After the initial wireframe, I create mockups that are like real webpages (hi-fidelity model) for these pages. Here are the screenshots of the final mockups, and comparison to the original ones: (Design Tool: UXPin)

Homepage

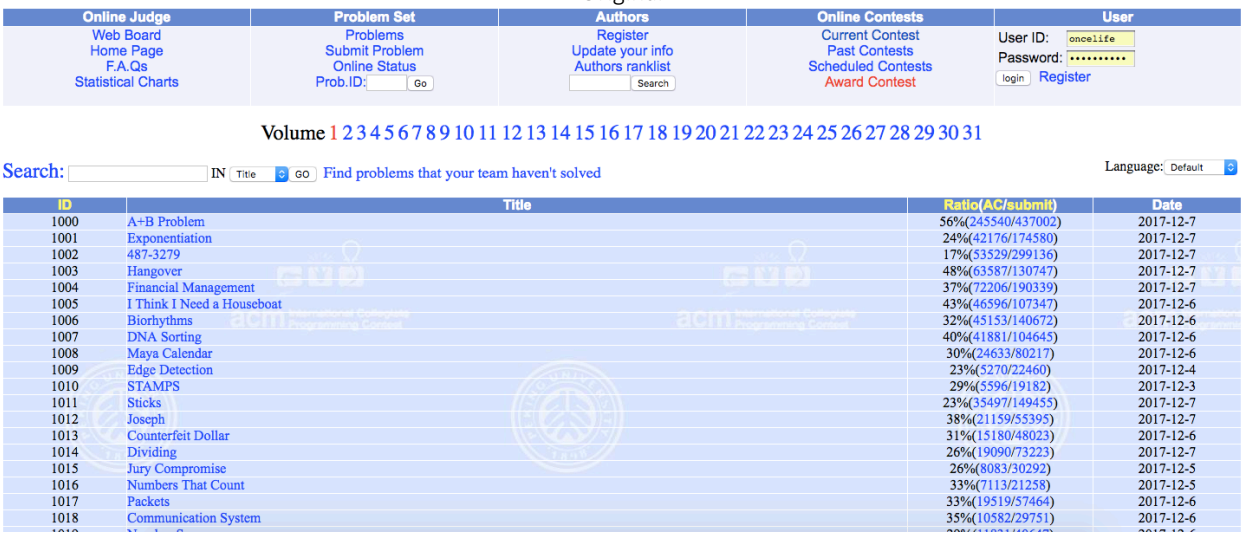
Original

Redesigned

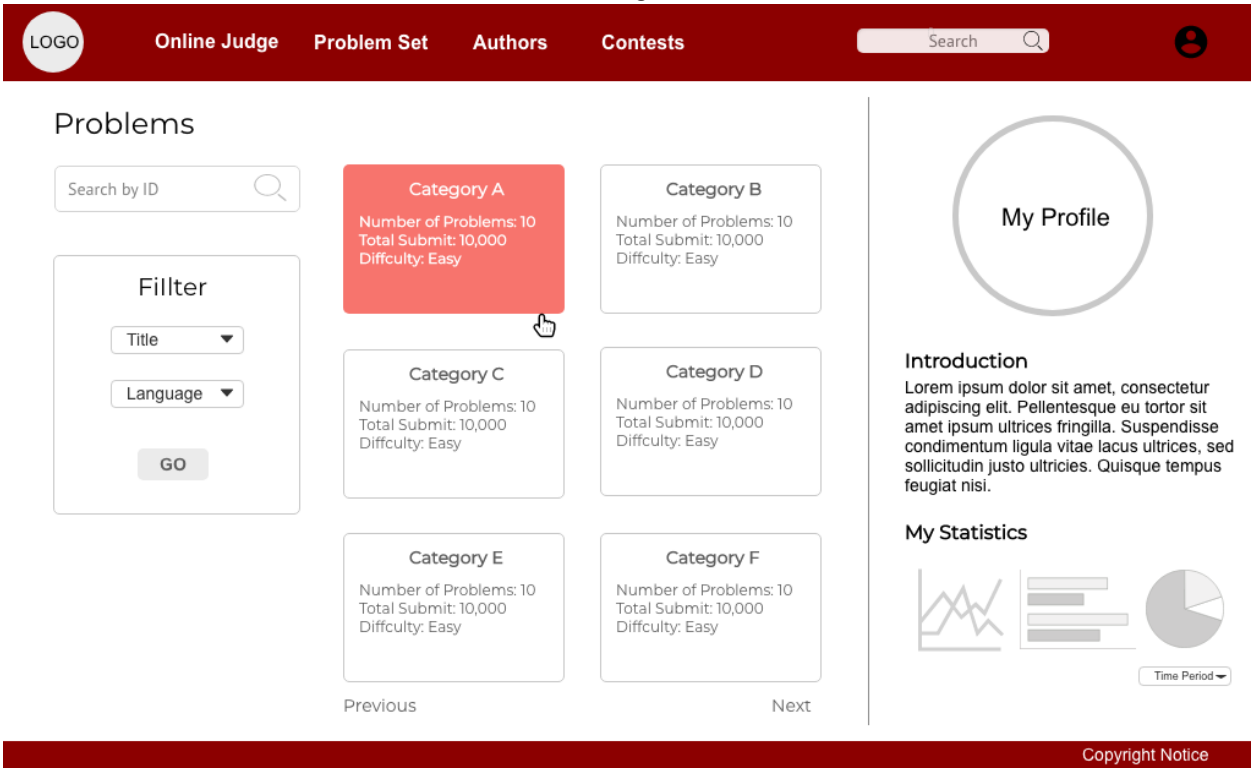


Problems set page

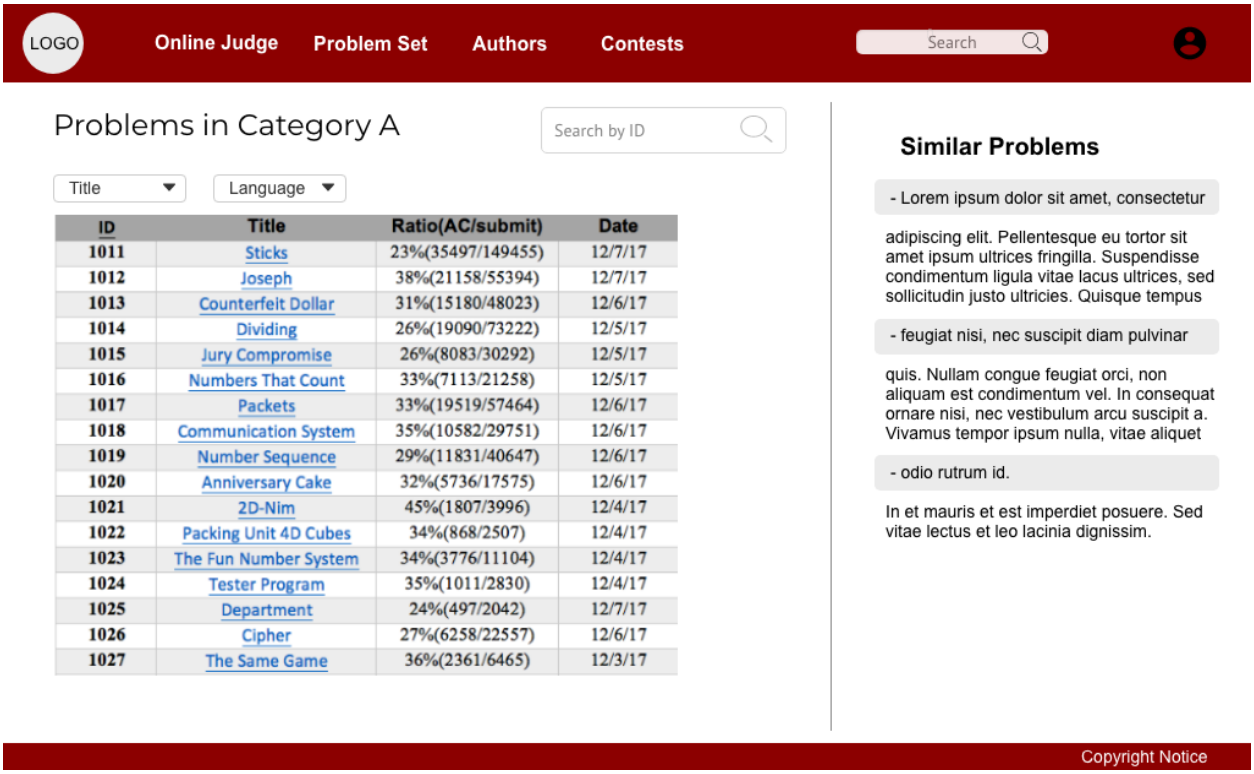
Original



Redesigned



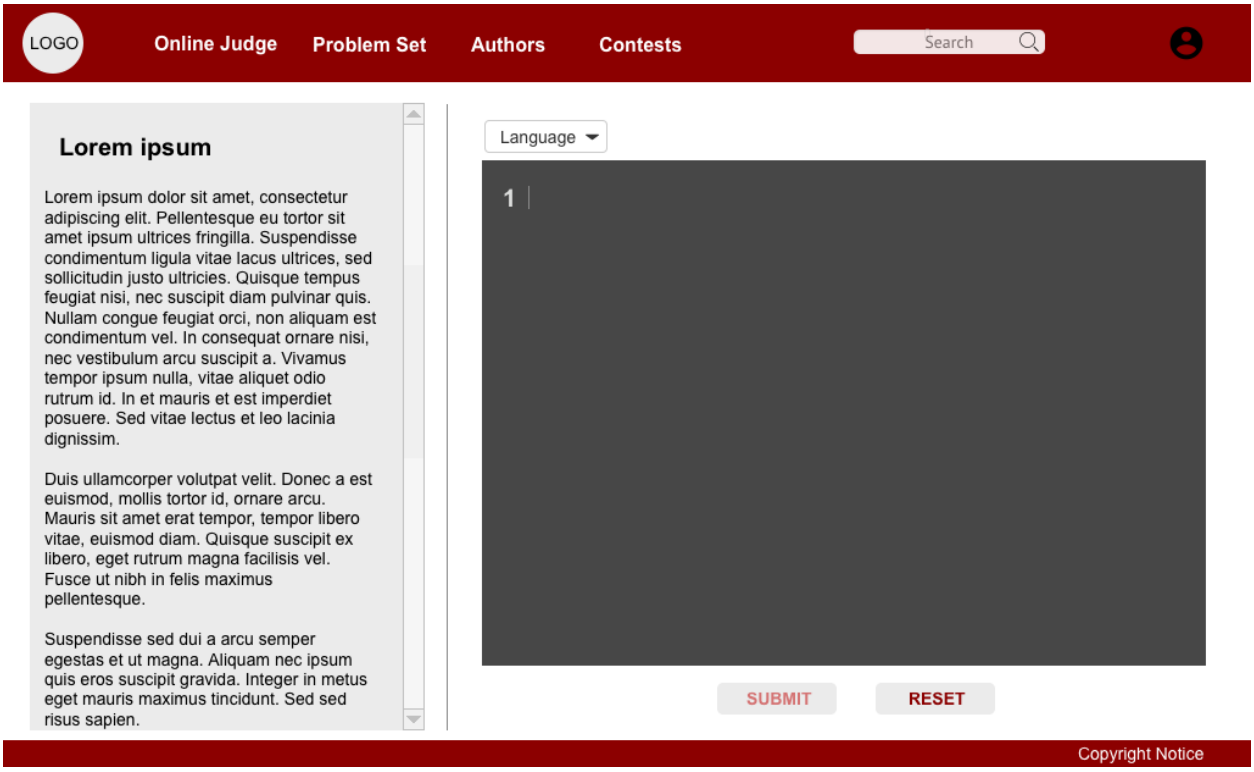
Detailed problems page (not present in original website)



Coding page

Original

Redesigned



Contest page

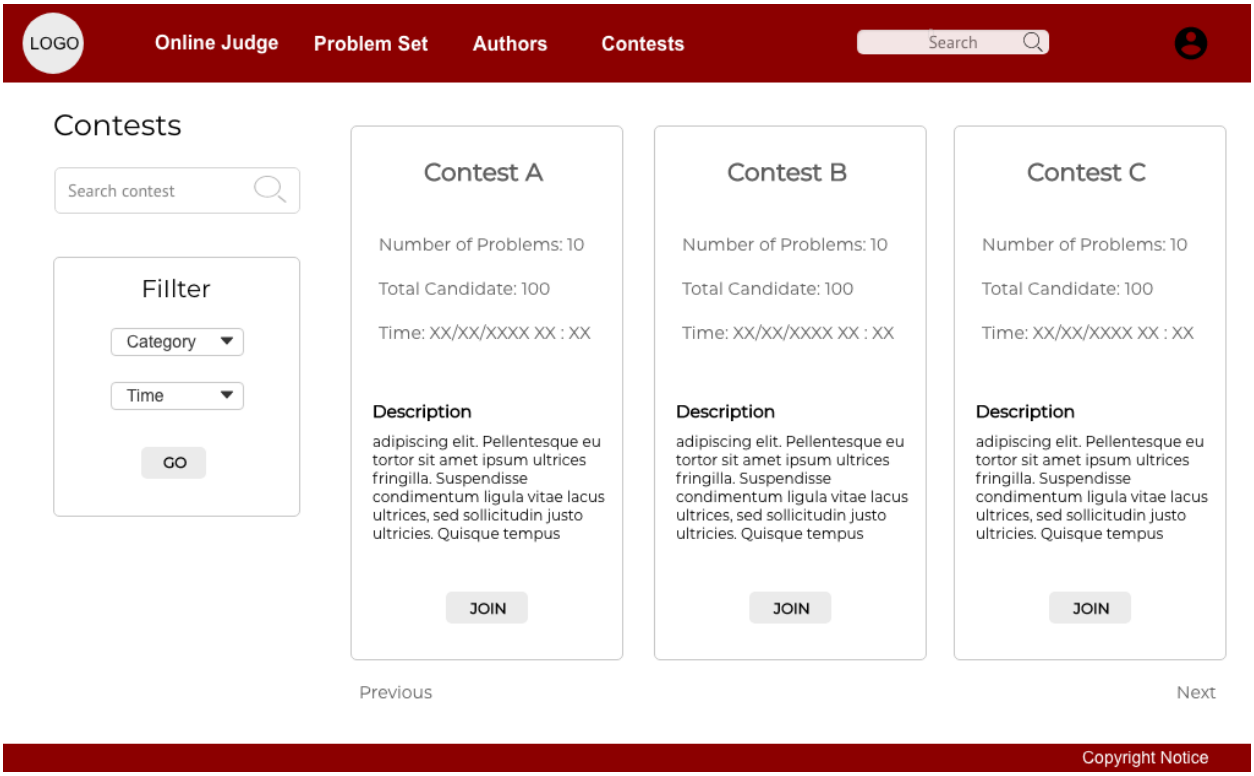
Original

Online Judge	Problem Set	Authors	Online Contests	User
Web Board	Problems	Register	Current Contest	oncelife Log Out
Home Page	Submit Problem	Update your info	Past Contests	Mail:0(0)
F.A.Qs	Online Status	Authors ranklist	Scheduled Contests	Login Log Archive
Statistical Charts	Prob.ID: Go	Search	Award Contest	

ID	Title	Start Time	Status
1528	POJ Challenge Round 5(绍兴一中邀请赛)(This contest is hosting in http://poj.openjudge.cn/challenge5/)	2014-01-26 18:30:00.0	ended
1527	POJ Challenge Round 4(雅礼中学邀请赛)(This contest is hosting in http://poj.openjudge.cn/challenge4/)	2013-12-22 14:00:00.0	ended
1526	POJ Challenge Round 3(This contest is hosting in http://poj.openjudge.cn/challenge3/)	2013-11-10 19:00:00.0	ended
1525	测试1	2013-10-12 14:50:00.0	ended
1524	POJ Challenge Round 2(长郡中学邀请赛)http://poj.openjudge.cn/	2013-10-13 19:00:00.0	ended
1523	POJ Challenge Test Round 1(This contest is hosting in poj.openjudge.cn)	2013-06-02 14:00:00.0	ended
1522	ACM Trainning	2012-10-06 00:00:00.0	ended
1521	IOI Training	2012-09-20 13:00:00.0	ended
1520	ACM Training 2012.7.27	2012-07-27 10:00:00.0	ended
1519	ACM Training 2012.7.26	2012-07-26 10:00:00.0	ended
1518	ACM Training 2012.7.25	2012-07-25 10:00:00.0	ended
1517	ACM Training 2012.7.24	2012-07-24 10:00:00.0	ended
1516	ACM Training 2012.7.23	2012-07-23 10:00:00.0	ended
1515	ACM Training 2012.7.22	2012-07-22 10:00:00.0	ended
1514	ACM Training 2012.7.20	2012-07-20 10:00:00.0	ended
1513	ACM Training 2012.7.19	2012-07-19 10:00:00.0	ended
1512	ACM Training 2012.7.18	2012-07-18 10:00:00.0	ended
1511	ACM Training 2012.7.17	2012-07-17 10:00:00.0	ended
1510	ACM Training 2012.7.16	2012-07-16 10:00:00.0	ended
1509	ACM Training 2012.7.15	2012-07-15 10:00:00.0	ended

[Top] [Previous Page] [Next Page]

Redesigned



Usability Improvements for the redesigned model

- Intuitive Design
  By hiding sub options into dropdowns and dividing each page into left, right parts, it will give out a more intuitive structure

- Ease of Learning
  To raise learnability, I change layout more similar to famous counterparts, and use different color to highlight important parts

- Efficiency of Use
  Every page is in a more clear and structured layout. For experienced users, it's easy to memorize the procedures

- Memorability
  Change list and table to graphical components, easier for memorizing

- Error Frequency and Severity
  Enough space between different items, also apply different colors to distinguish. Error rate is expected to be lower

- Subjective Satisfaction
  After the redesign, the website now has a clear structure with good learnability. Thus, most users will be satisfied using it

Tradeoffs

There are certain tradeoffs I made in achieving those goals. A significant one is the adding of large amount of new information, like problem categories, contest candidate numbers, user's own statistics, which need to be calculated

when actually implementing the website. Another one is the reduce in maximum information shown in one page, as now the gap between different blocks largely increases.

VI. Addition Task: Programming one page

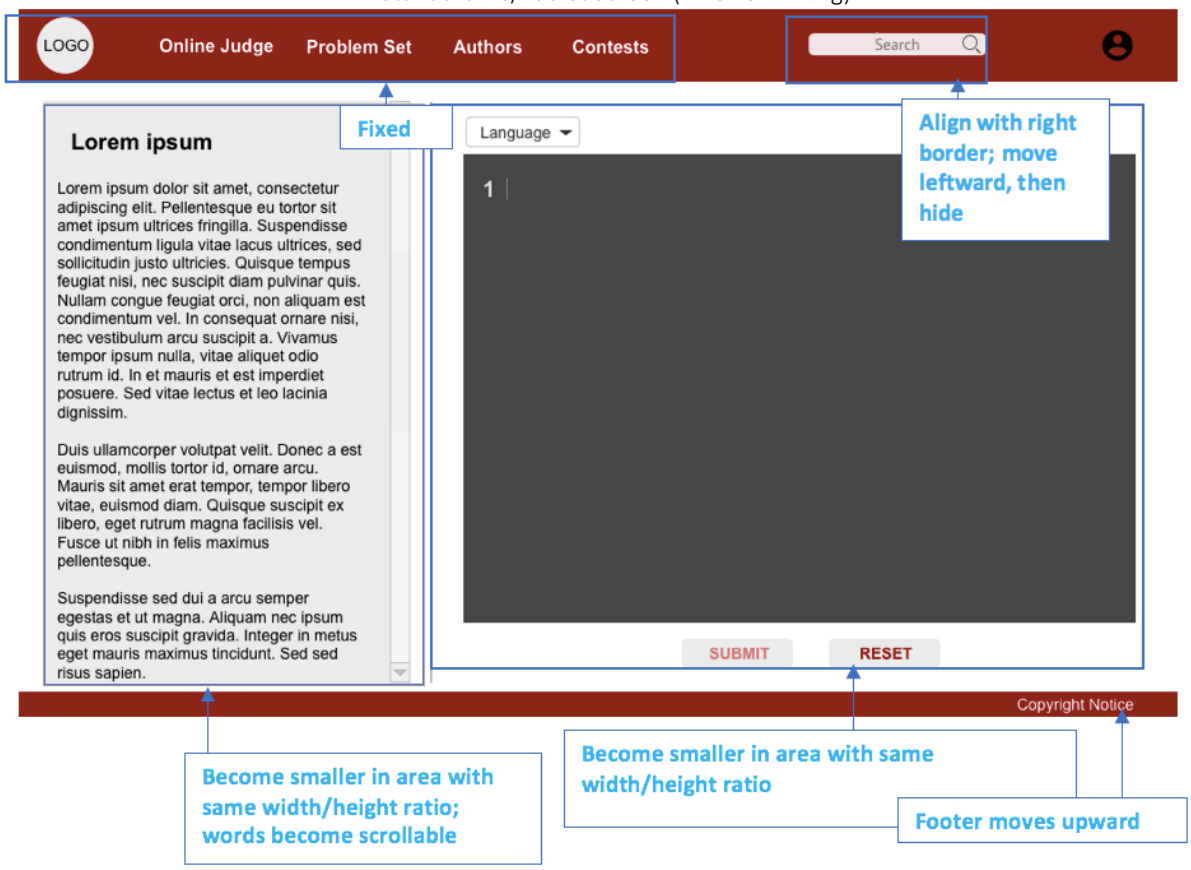
After finishing the visual redesign, I decide to technically implement one page to give a better idea on how it actually looks like. The page I chose to make is the coding page, as this is the page that serves core function of the whole website.

Writing responsive pages is a fundamental idea in modern web app development. First, I will show the responsiveness of this page on different screens. One way to easily tell the difference is by adding annotations to the hi-fi mockup.

Annotated mockup

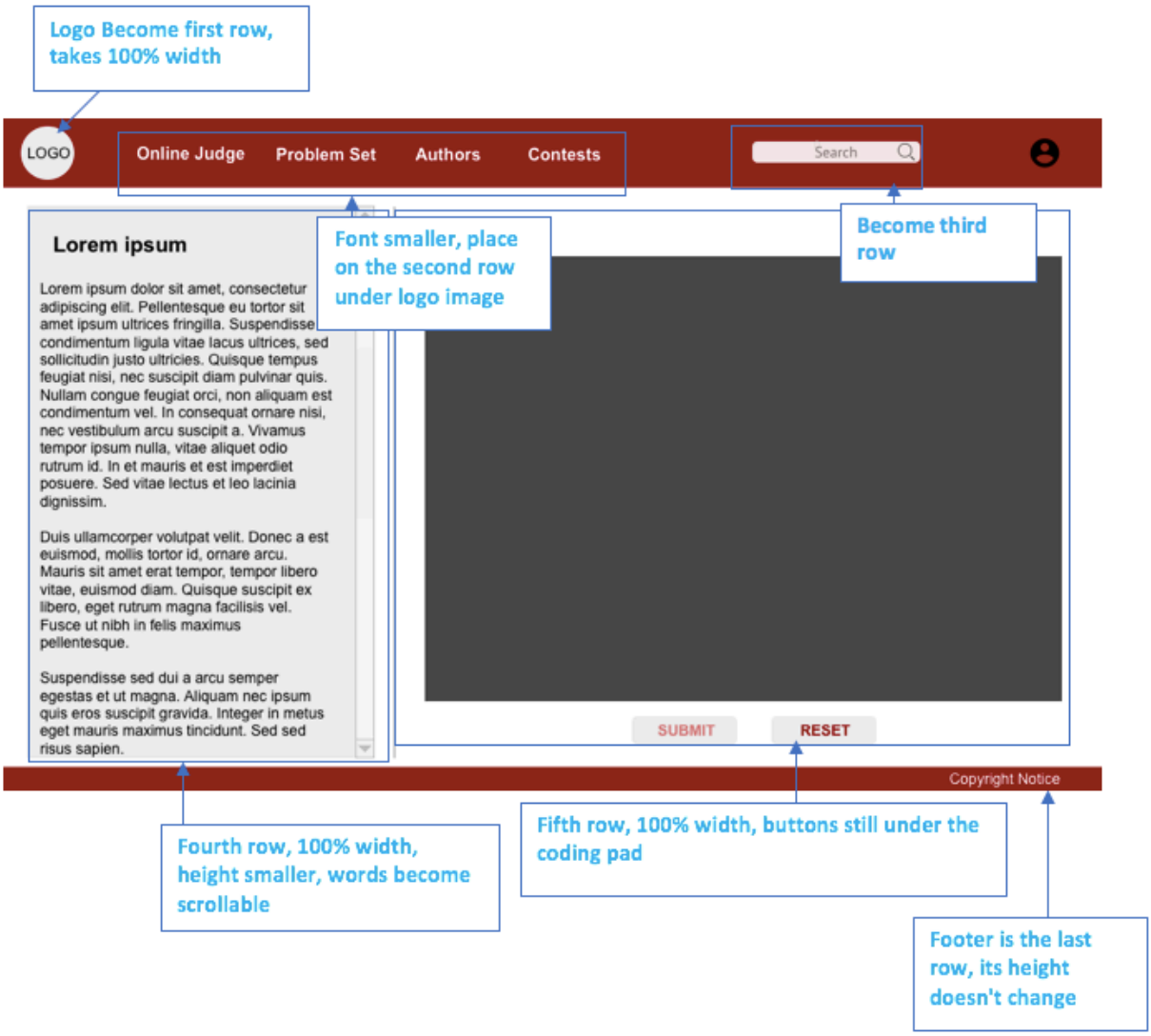
Due to its programming website property, the primary use of this website would be on PC/Tablet with keyboard:

Standard PC/Tablet Screen (when shrinking)



However, with the continuous growing usage of smart phone, a page that automatically fits into phone screens should also be considered. I add a media query on 600px width, to change the page to phone mode:

Change to phone mode (When width <= 600px)



Coding the page

I write this in standard HTML/CSS format. For responsiveness handling, I choose CSS Grid, a method which divide page into grids and automatically handles relevant responsive changes. The final work includes one HTML file (index.html), one CSS file (style.css) and an image folder.

View the page at: https://edwardcgeorge.github.io/poj-redesign/