1. **Customer Statement of Requirements (CSR)**

**1.1 Problem Statement**

There are many websites that offer programming challenges. Typically, these sites are used by an interviewee preparing for a job interview, or a computer science student furthering their education. These sites vary in usability and focus. If you are looking to practice and hone your programming skills across a wide variety of topics for potential upcoming interviews or to help study for upcoming classes, you face a variety of problems. A typical sequence if you are looking for an appropriate challenge or challenges is as follows:

1. **Navigate the web to an appropriate site**
2. **Learn the site U.I. and navigation**
3. **Find a challenge that suits your interests**
4. **Repeat 1-3 until sufficiently prepared**

This current process leaves a lot to be desired and that forces the user to spend an unnecessary amount of time on web browsing and learning various site navigation. The potential solution should be one that eliminates the tedium that comes from having to repeat steps one and two of this process and save the users valuable time that they could use to prepare for an interview or an upcoming test.

**1.2 Solution**

The proposed solution should present these challenges from multiple sites and present them in a unified design that has a centralized location. Ideally the new sequence should be:

1. **Download/Access the app/site**
2. **Learn the site U.I. and navigation**
3. **Find a challenge that suits your interests**
4. **Repeat 3 until sufficiently prepared**

Users should no longer have to find challenge sites on their own, and they will be able to view challenges from other sites without having to learn a new interface. Thus, eliminating the repletion in steps one and two and ultimately saving the user time. Finally, by pulling from many sources the user will be offered with the broadest possible selection of challenges that they can find at a single location.

The proposed solution must be able to gather and store adequate information about each challenge. This information should be reliable, and a user should not have to worry about a challenge “disappearing” and be forced to scour the web. It should gather enough information that a user should be able to complete a challenge without the need of going to the source website, however they should be able to find the source website if desired.

To ensure that users can adequately locate and complete challenges, a serviceable interface must be provided so they can find challenges to their liking. Showing every problem from all the included sites could overwhelm potential users. Users must be able to use a menu to search and filter the database to limit the quantity of entries to a more manageable form. They may only be interested in problems from a single or several source sites, or only looking for problems of a easy difficulty. Finally, the results must be presented in a way that users can view and find the complete text and other challenge specific information.

A secondary goal of the project should be a more personalized user experience. This includes allowing the user to upload their own personal challenges. Also, a user may want to be to create a list of favorite challenges and save them for later. Users may want to be able to toggle the status of challenges between solved, viewed but not solved, not viewed, skipped, and newly downloaded. A user may wish to add text or attach files associated with the challenges, so they can revisit their implementations or, so they can review past solutions. These functions would allow a user a unique view of the site and provide additional advantages if they do continually use the product over time.

The only true way that this project solves the problem if a user finds it easier to locate challenges than manually going each site. Great care must be taken when creating the interface. The time to learn the created interface must be significantly lower than the time to learn the sum of the websites interfaces. Additionally, if searching the database is slow and unreliable, then the original process of visiting each site might be more convenient to the user. It is not enough to simply present the user with access to all challenges. A user **must** save time when comparing with the traditional method to be a true solution.

1. **Glossary of Terms**

**Challenge**- A programming challenge commonly found on the internet or in computer science coursework. Typically focuses on learning a specific concept over practical application.

**User-** A person that is using the application.

**Database –** the storage of all the collected challenges and text associated with the challenges and accompanying websites.

**Web Scraper-** A program that gathers publicly available data from the internet and typically stores it in a database.

**User Interface-** The interface that lets the user interact with and search within the database.

**Search/Query-** A user filters or limits the results using certain parameters such as site origin.

**Results-** The list of challenges that is presented to the user after a search query.