A

Project Report

On

**Code forces Problem Picker**

[**https://drive.google.com/drive/folders/1F-oKQzGCx\_4WhkI77OE07aYzwvVSvpok?usp=sharing**](https://drive.google.com/drive/folders/1F-oKQzGCx_4WhkI77OE07aYzwvVSvpok?usp=sharing)

Submitted in partial fulfillment of the requirement for the VI semester

**Bachelor of Technology**

By

Kritik Manral- 1961087

**Under the Guidance of**

**Mr. Shobhit Kumar**

**Assistant Professor**

**Dept. of CSE**



**DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS**

**SATTAL ROAD, P.O. BHOWALI,**

**DISTRICT- NAINITAL-263132**

**2021- 2022**

**STUDENT’S DECLARATION**

We hereby declare the work, which is being presented in the project, entitled “**Codeforces Problem Picker**“ in partial fulfillment of the requirement for the award of the degree **B.TECH**  in the session **2021-2022**, is an authentic record of my own work carried out under the supervision of

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date:02/07/2022 Kritik Manral

**CERTIFICATE**

**The project report entitled “Codeforces Problem Picker” being submitted by** Kritik Manral**,** Chirag Lohani & Pranay Sah **enrollment no.** PV-B1961087, PV-B1961035 & PV-B1961128 reapectively **to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. They have worked under my guidance and supervision and fulfilled the requirement for the submission of report.**

**Mr. Shobhit Kumar Dr. Ankur Bisht**

**(Project Guide ) (HOD, CSE Dept.)**

**ACKNOWLEDGEMENT**

I take immense pleasure in thanking Honorable **“Mr. Shobhit Kumar”** to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me everything that I need. I again want to extend thanks to our President **“Prof. (Dr.) Kamal Ghanshala”** for providing us all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to Professor **“Dr. Ankur Bisht”** (HOD-CS&A, GEHU) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this thesis.

Finally, yet importantly, We would like to express our heartiest thanks to our beloved parents, our brother and sister for their moral support, affection and blessings. We would also like to pay our sincere thanks to all our friends and well-wishers for their help and wishes for the successful completion of this research.

**Kritik Manral** [**kritikmanral1@gmail.com**](mailto:kritikmanral1@gmail.com)

**ABSTRACT**

This project is about an Codeforces Extension to get particular user’s unsubmitted questions. This project is a prototype for the Codeforces Extension and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the all the Competitive Programmers giving contest and practicing on Codeforces. This project uses the Codeforces API to get the required information about a user profile submission and respective wrong answers.

In addition to that, The Codeforces Extension main use case is to fetch the details from the codeforces.com using the API calls and just use that details to fetch the required information and then display it in a GUI manner.It has all the require HTML CSS JAVASCRIPT files to attain the required purpose in order to show the relevant details using the API calls in a eye catching GUI manner with all the details dynamic to be redirected to the corresponding links.

In a nutshell, the methodology for developing this project is the agile development cycle. The agile development cycle consists of six phases, which is requirements analysis, planning, design, implementation or development, testing, and deployment. Due to the iterative and flexible nature of this approach, it is able to effectively adapt to users with changing requirements.

**Table Of Contents**

**Declaration I**

**Certificate II**

**Acknowledgement III**

**Abstract IV**

**Table of Contents VI**

**CHAPTER1 INTRODUCTION**

1.1 Prologue

1.2 Background and Motivations

1.3 Problem Statement………………………………………………

1.4 Objectives

**CHAPTER 2: PROPOSED SYSTEM**

2.1 History

**CHAPTER 3: S/W AND H/W REQUIREMENTS**

**CHAPTER 4: DFD**

**CHAPTER 5: CODING OF FUNCTION**

**CHAPTER 6:** **TESTING STRATEGIES**

**CHAPTER 7: LIMITATIONS**

**CHAPTER 8: ENHANCEMENTS**

**CHAPTER 9: CONCLUSION**

**REFERENCES**

1. **Introduction**
   1. **Prologue**

Codeforces is a website that hosts competitive programming contests.It is maintained by a group of competitive programmers from ITMO University led by Mikhail Mirzayanov. Since 2013, Codeforces claims to surpass Topcoder in terms of active contestants. As of 2018, it has over 600,000 registered users. Codeforces along with other similar websites are used by top sport programmers like Gennady Korotkevich, Petr Mitrichev, Benjamin Qi and Makoto Soejima, and by other programmers interested in furthering their careers.

**Academic use**

Codeforces is recommended by many universities. According to Daniel Sleator, professor of Computer Science at Carnegie Mellon University, competitive programming is valuable in computer science education, because competitors learn to adapt classic algorithms to new problems, thereby improving their understanding of algorithmic concepts. He has used Codeforces problems in his class, 15-295: Competition Programming and Problem Solving.

The purpose of this document is to build an Codeforces Extension to get particular user’s unsubmitted questions.

1.1.2 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the Codeforces Extension and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the all the Competitive Programmers giving contest and practicing on Codeforces.

1.1.3 PROJECT SCOPE

The purpose of the extension is to ease to fetch the details of the previously unsolved questions on Codeforces site and according to the lowest rating suggest the user to retry the question which he/she wasn’t able to solve during mid contest or during practice

This project uses the Codeforces API to get the required information about a user profile submission and respective wrong answers.

* 1. **Background and Motivation**

Whenever we friends use to do coding in codeforces we use to discuss the problems we were unable to do. However when we were not in university or together it use to become difficult discussing what questions we were doing and what question we are unable to do or they are not done completely by our friends or even our seniors.

So we decided to make a project that can give us the questions that are difficult and our friends or any particular user is unable to complete it.

Our project can give the name and no. of the question. We just need to type the user name of the person we intent to check

* 1. **Problem Statement**

The problem domains on this project are:

1. **No existing similar website-** The website that can fetch the details of the previously unsolved questions on Codeforces site and according to the lowest rating suggest the user to retry the question which he/she wasn’t able to solve during mid contest or during practice.

This project uses the Codeforces API to get the required information about a user profile submission and respective wrong answers.

1. **Make simple, attractive and easy to use website –** User wanted a easy to use website with no fuss and unecessary details. Just aim should to stick on the main aim which was to get the unsolved questions.
   1. **Objectives**

The objective of this is to:

**1.Make it with a simple feature and run smoothly** By using this the user will get the direct access to the question which he wanted to see. With clean and visually attractive UI the user will get a good experience. The website should not contain any ads which will enhance the user experience even more.

**2.Support quick search** searching should be fast, so that users time no not get wasted.

1. **Proposed System**
   1. **History**

Codeforces was created by a group of competitive programmers from [Saratov State University](https://en.wikipedia.org/wiki/Saratov_State_University) led by Mike Mirzayanov. It was originally created for those interested in solving tasks and taking part in competitions. The first Codeforces Round was held on the February 19, 2010 with 175 participants. As of the end of July 2019 over 650 rounds were held, with over 9000 registered competitors per round on average. Before 2012 Codeforces Rounds were titled "Codeforces Beta Rounds" to indicate that the system was still under development.

[Codeforces API](https://codeforces.com/apiHelp) was introduced over 7 years ago . Since then, not much changed in the API itself, but there were a lot of projects done with it.

I would like to present [my attempt](https://github.com/Covariance/codeforces-api) to make a Codeforces API wrapper for Java. As it can be clearly seen while browsing GitHub, there're some takes on making it but none of them is finished (believe me, I've checked). Many amazing things can be done via Codeforces API, such as [this app](https://apps.apple.com/us/app/codeforces-watchr-contests/id1495591299), [this site](https://cfviz.netlify.app/), or [this app](https://github.com/Abhijeet-AR/Competitive_Programming_Score_API).

And there are a lot of other projects that can be done via write methods API that are currently done by page parsing, such as all sorts of CLIs for Codeforces. Hope that in some time in the future they would be added to the official API, so the development of such applications would become much easier.

**SYSTEM FEATURES**

* **DESCRIPTION and PRIORITY**

The extension should maintain all the records of the users submission and should update accordingly as per submissions.

* **STIMULUS/RESPONSE SEQUENCES**
  + Redirect to the actual question link on clicking the question.
  + Displays a detailed list of unsolved question and give “recommendations” accordingly.
  + Sort all the question as per the ratings.

API

JS FILE

USER WINDOW

FRONT\_END

1. **Hardware and Software required**

Operating environment for the Codeforces Extension is as listed below.

It will be a cloud based extension that can be coupled with any of the modern browser as an external 3rd-party extension.

* Microsoft Edge
* Google Chrome
* Safari Browser
* Mozilla Firefox

**USER INTERFACES**

* Front-end software: HTML CSS JAVASCRIPT
* Back-end software: NODE JS

**HARDWARE**

* Computer system.
* Monitor
* Keyboard
* Independent of OS.
* A browser which supports CGI, HTML & Javascript.

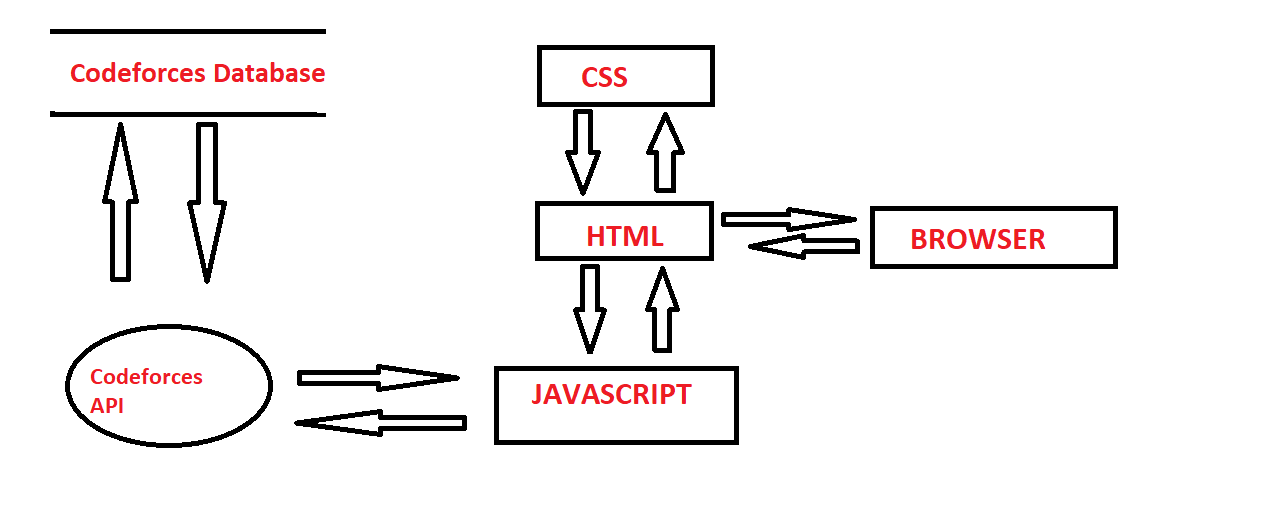
**SOFTWARE**

Following are the software used for the Codeforces Extension

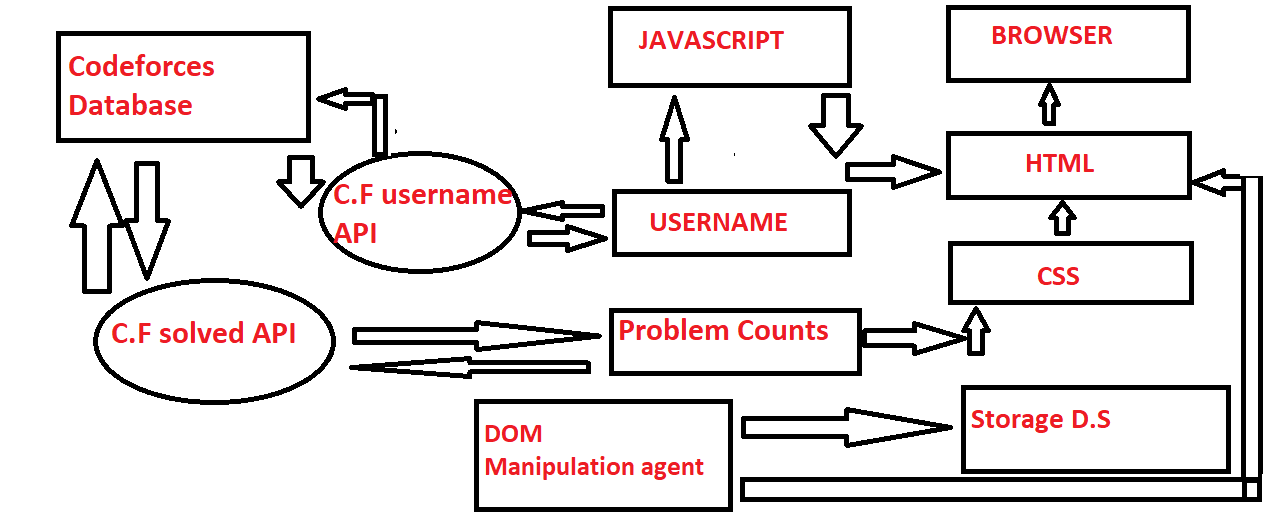
|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | INDEPENDENT. |
| API | To get user data. |
| Front-end | To show the details in a window. |
| Back-end | To handle the response from the APIs |

1. **DFD (DATA FLOW DIAGRAM)**

**LEVEL 0 DFD-**

****

**LEVEL 1 DFD-**

****

1. **Testing Strategies**

**There are different testing strategies for our website**

**1. Functionality Testing**

**Functionality Testing of a Website** is a process that includes several testing parameters like user interface, APIs, database testing, security testing, client and server testing and basic website functionalities. Functional testing is very convenient and it allows users to perform both manual and automated testing. It is performed to test the functionalities of each feature on the website.

Web based Testing Activities includes:

**Test**all **links**in your webpages are working correctly and make sure there are no broken links. Links to be checked will include –

* Outgoing links
* Internal links
* Anchor Links
* MailTo Links

**Test Forms**are working as expected. This will include-

* Scripting checks on the form are working as expected. For example- if a user does not fill a mandatory field in a form an error message is shown.
* Check default values are being populated
* Once submitted, the data in the forms is submitted to a live database or is linked to a working email address
* Forms are optimally formatted for better readability

**Test Cookies** are working as expected. Cookies are small files used by websites to primarily remember active user sessions so you do not need to log in every time you visit a website. Cookie Testing will include

* Testing cookies (sessions) are deleted either when cache is cleared or when they reach their expiry.
* Delete cookies (sessions) and test that login credentials are asked for when you next visit the site.

**Test HTML and CSS** to ensure that search engines can crawl your site easily. This will include

* Checking for Syntax Errors
* Readable Color Schemas
* Standard Compliance. Ensure standards such W3C, OASIS, IETF, ISO, ECMA, or WS-I are followed.

**Test business workflow**– This will include

* Testing your end – to – end workflow/ business scenarios which takes the user through a series of webpages to complete.
* Test negative scenarios as well, such that when a user executes an unexpected step, appropriate error message or help is shown in your web application.

**Tools that can be used**: [**QTP**](https://www.guru99.com/quick-test-professional-qtp-tutorial.html) , IBM Rational , [**Selenium**](https://www.guru99.com/selenium-tutorial.html)

**2. Usability Testing**

[Usability Testing](https://www.guru99.com/usability-testing-tutorial.html) has now become a vital part of any web based project. It can be **carried out by testers** like you **or a small focus group** similar to the target audience of the web application.

**Test**the site **Navigation**:

* Menus, buttons or Links to different pages on your site should be easily visible and consistent on all webpages

**Test**the **Content**:

* Content should be legible with no spelling or grammatical errors.
* Images if present should contain an “alt” text

**Tools that can be used**: Chalkmark, Clicktale, Clixpy and Feedback Army

**3. Web UI Testing**

One of the most important interfaces within a web application are web server and application server interface and database server interface. [Web UI testing](https://www.testing-whiz.com/web-ui-comparison-and-functional-test-automation) will ensure that all the individual components within a web application are connected appropriately. One should check whether the interaction between these servers are executed properly or not with the help of this testing method.

**4. Compatibility Testing**

Compatibility tests ensures that your web application displays correctly across different devices. This would include-

**Browser Compatibility Test**: Same website in different browsers will display differently. You need to test if your web application is being displayed correctly across browsers, JavaScript, AJAX and authentication is working fine. You may also check for[Mobile](https://www.guru99.com/mobile-testing.html)Browser Compatibility.

The rendering of web elements like buttons, text fields etc. changes with change in **Operating System**. Make sure your website works fine for various combination of Operating systems such as Windows, Linux, Mac and Browsers such as Firefox, Internet Explorer, Safari etc.

**Tools that can be used**: NetMechanic

**5. Performance Testing**

Performance testing will help you determine the performance of your web application under various scenarios. Performance testing usually involves stress testing, scalability testing and load testing. In this testing method, website is usually tested for its functionality on different operating system, hardware platforms and more.

**6.Interface Testing:**

Three areas to be tested here are – Application, Web and Database Server

* **Application:** Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.
* **Web Server**: Test Web server is handling all application requests without any service denial.
* **Database Server:**Make sure queries sent to the database give expected results.

**Test system response** when **connection between the three layers**(Application, Web and Database) **cannot be established** and appropriate message is shown to the end user.

**Tools that can be used**: AlertFox, Ranorex

**Limitations**

* We have not stored the data of the user in our database.
* User cannot see when the date and time when those question were done.
* User should know the user ID of the person u need to check.

1. **Enhancement**

* We can add database to store the data of the user using our website and the no. of questions he attempts from our website.
* We can add more features in the website like we can search along with the tags of questions.

1. **Conclusion**

In this report, a functional website of “Codeforces Probem Picker” based on codeforces API and javascript have been developed for doing unfinished questions of other users in codeforces application. Project include searching of questions after logging in to our website with their codeforces login id and then searching the questions using the username of the person to serach.

**References**

* [https://codeforces.com](https://krazytech.com/projects)
* <https://en.wikipedia.org/wiki/Codeforces>
* WHATWG (Web HyperText Application Working Group) HTML Specfication @ https://html.spec.whatwg.org/multipage/.
* HTML 4.01 Specification W3C Recommendation 24 December 1999 (@ https://www.w3.org/TR/html401/).
* XHTML 1.0 Specification W3C Recommendated Revised 1 August 2002 (@ https://www.w3.org/TR/xhtml1/).
* CSS 2.1 Specification W3C Recommendation Revised 17 December 2014 (@ https://www.w3.org/TR/CSS21/).
* W3C CSS Specification @ https://www.w3.org/Style/CSS/specs.en.html.
* ECMA-262 ECMAScript (JavaScript) Specification @ https://www.ecma-international.org/publications-and-standards/standards/ecma-262/.
* Mozilla's MDN JavaScript Reference @ https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
* jQuery mother site @ https://jquery.com/ and jQuery API @ https://api.jquery.com/.