ONESTOP CODING DASHBOARD USING WEB SCRAPING

A Project Report submitted in partial fulfillment of the requirements for the award

of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Bhange Ravi Shankar, 121810311001

Kasa Prathyush Kumar, 121810311036

Madarapu SNSSM Meghana, 121810311044

Seelam Sowmith, 121810311049

Under the esteemed guidance of

Dr.S.V.Lakshmi

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM

(Deemed to be University)

VISAKHAPATNAM

DECEMBER 2021-APRIL 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

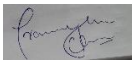
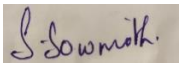
GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the project review entitled **"ONESTOP CODING DASHBOARD USING WEB SCRAPING"** is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in the Computer science and Engineering. The work has not been submitted to any other college or university for the award of any degree or diploma.

Date: 05-04-2022

Registration No(s).	**Name(s)**	**Signature(s)**
121810311001	Bhange Ravi Shankar	
121810311036	Kasa Prathyush Kumar	
121810311044	Madarapu SNSSM Meghana	
121810311049	Seelam Sowmith	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled **"ONESTOP CODING DASHBOARD USING WEB SCRAPING"** is a bonafide record of work carried out by **Bhange Ravi Shankar,121810311001; Kasa Prathyush Kumar,121810311036; Madarapu SNSSM Meghana.121810311044; Seelam Sowmith,121810311049** students in partial fulfillment of requirements for the award of the degree of Bachelor of Technology in the Computer Science and Engineering.

Project Guide

Dr. S.N.V. Jitendra M

Assistant Professor

Head of the department

Dr. R. Sireesha

Professor

TABLE OF CONTENTS

1.	Abstract		01
2.	Introduction		02
3.	Literature Review		03
4.	Problem Identification & Objectives		04
5.	System Methodology		05-06
6.	Overview of Technologies		07-08
7.	Implementation		09-21
	7.1	Coding	
	7.2	Testing	
8.	Results & Discussions		22-28
9.	Conclusion & Future Scope		29
10.	References		30

1. ABSTRACT

As you know, it is difficult for users to navigate each website. This website creates users' coding portfolios and provides user activity which includes his/her no of problems solved in a day. By taking scores from different coding websites, we create the overall score and provide the leaderboard with global ranking and institute ranking which makes users analyze where his/her position is. For calculating the overall score of a user, we used a standard mathematical equation that gives an accurate score according to his/her individual scores on individual platforms. The website also shows the daily activity status. And coming to the activity of a user will be tracked every 24 hours. It is implemented by creating a background thread that runs and performs tasks like updating user scores if anything changed. We considered websites like Codechef, Codeforces, GeeksforGeeks, and InterviewBit. We retrieved users' data like no of problems solved, rating, score, streak. For retrieving data from websites, we used web scraping technology. Backend is designed using Java 11 and for web scraping, we used JSOUP API.

2. INTRODUCTION

There are great platforms such as CODECHEF, CODEFORCES, GEEKSFORGEEKS, and INTERVIEWBIT, but it is very complicated for students and users to navigate from each platform to see scores, ratings, and upcoming contests. Our project is a website that can track all coding results from these platforms, provide an overall score, and maintain users' daily and weekly activity regardless of whether the user is active or not. It is designed to solve this kind of navigation problem faced by students/users when there is only one, whether to join a particular platform.

This website creates a user coding portfolio. By collecting scores from different coding platforms, we summarize the overall score and provide the leaderboard with a global and institutional ranking. This allows users to analyze their position. There is a contest page where you can see upcoming contests from various platforms such as CODECHEF, CODEFORCES, GEEKSFORGEEKS, and INTERVIEWBIT. This website also shows your daily activity status, such as the number of problems solved in one day and your weekly activity status. Data is retrieved from the live websites and stored in our database to track user activity and we are using JSOUP API for web scraping. User activity is updated every 24 hours.

We mainly focus on the overall score and activity of a user. To calculate the overall score of a user, we gave priority to contest rating and then problems solved in each platform. Graph of overall score of a user mostly depends on contest ratings of different websites. As his/her contest rating increases, the overall score also increases in the same way as a quadratic equation where contest ratings are real numbers and problems solved in each platform are considered constants. We used threads for each user to increase the performance of our web application while updating user activity data in the database.

3. LITERATURE REVIEW

The student performance tracker tool is very important to track his/her all performances done in their respective challenges or test. Here we are going to derive an efficient way to track student coding performance. Student coding performance can be calculated in many different ways but here we are concentrating on some factors to measure student performance. A pr study says that student performance is dependent on major factors like motivation, learning, experience, comfort, confidence level, etc. Our project includes these kinds of factors like motivation and confidence level. Overall problem solved by a student gives a kind of motivation and confidence level to students which helps them a lot to continue with their performance. Not only this but also overall score with global and institute-wise ranking gives us much better motivation to a student. In this project, we mainly focus on student coding performance in which he/she improves their coding experience and our project keeps track of all the no of problems solved in a week and the total no of problems solved in each website. Along with this we also add some features like viewing all contests in our page. We mainly focused on integrating four coding platforms and tracking the student performance by giving them overall scores and tracking their performance.

4. PROBLEM IDENTIFICATION AND OBJECTIVES

PROBLEM STATEMENT

Users have accounts on different coding platforms and each website has coding scores and contest ratings. But it's difficult to analyze the overall coding score. There will be various contests held on various platforms like CODECHEF, CODEFORCES, GEEKSFORGEEKS and INTERVIEWBIT. It's quite difficult to navigate to each website and check which contest is being held and when.

OBJECTIVES

Our objective is to make a single platform by integrating all four platforms (Codechef, Codeforces, GFG, InterviewBit) so that

- Users can view their all scores in one place
- User can have overall score and leaderboard
- Can check which contest is held at which platform
- Can check user daily activity and weekly activity

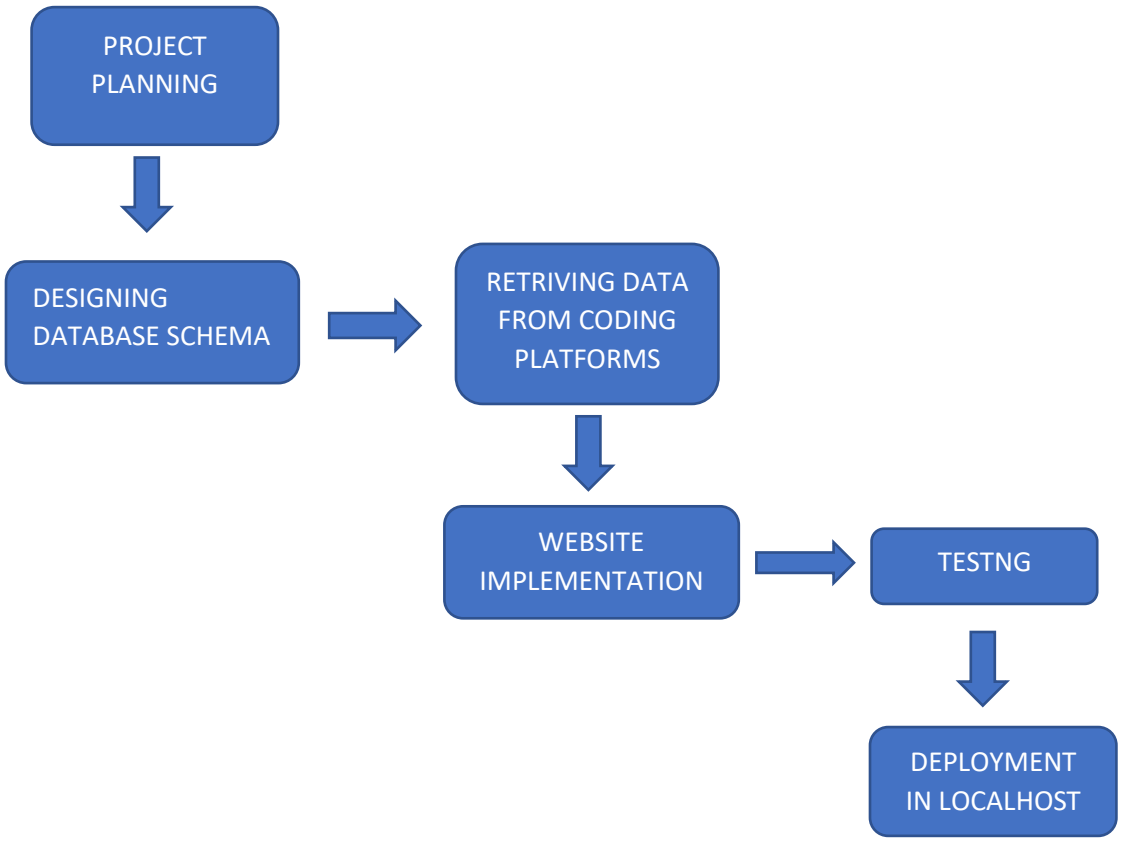
EXISTING SYSTEM

- We already have some mobiles apps like CodeClock, CPing, etc. But these apps are built for just notifying which contest is held on which platform and at what time and date. We will get notifications for every contest held on different platforms.

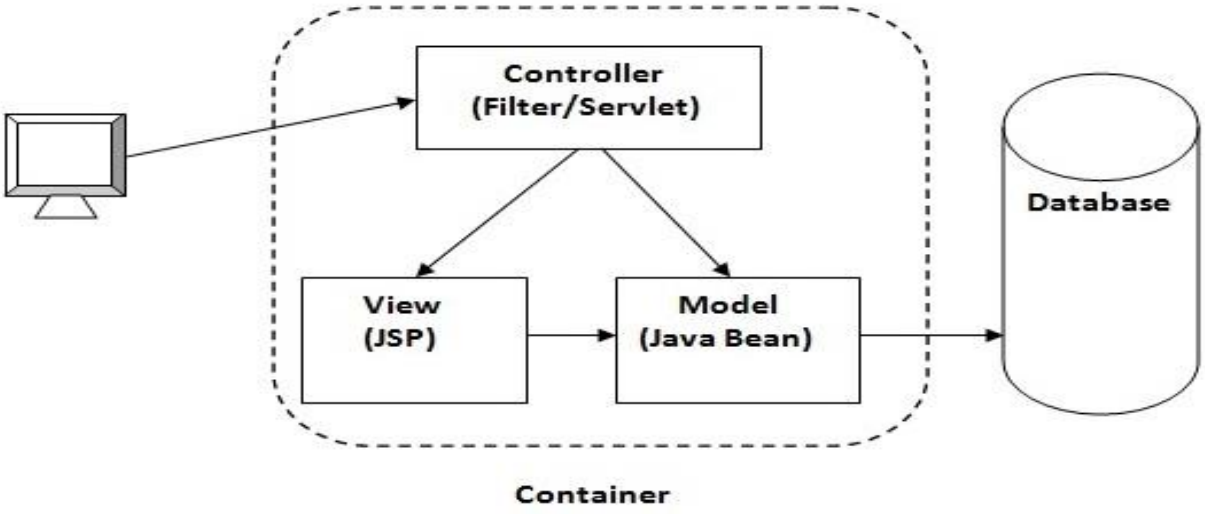
PROPOSED SYSTEM

- Along with the existing features, users can also check is activity and we proposed features like dashboards for each platform and overall score with leaderboards.

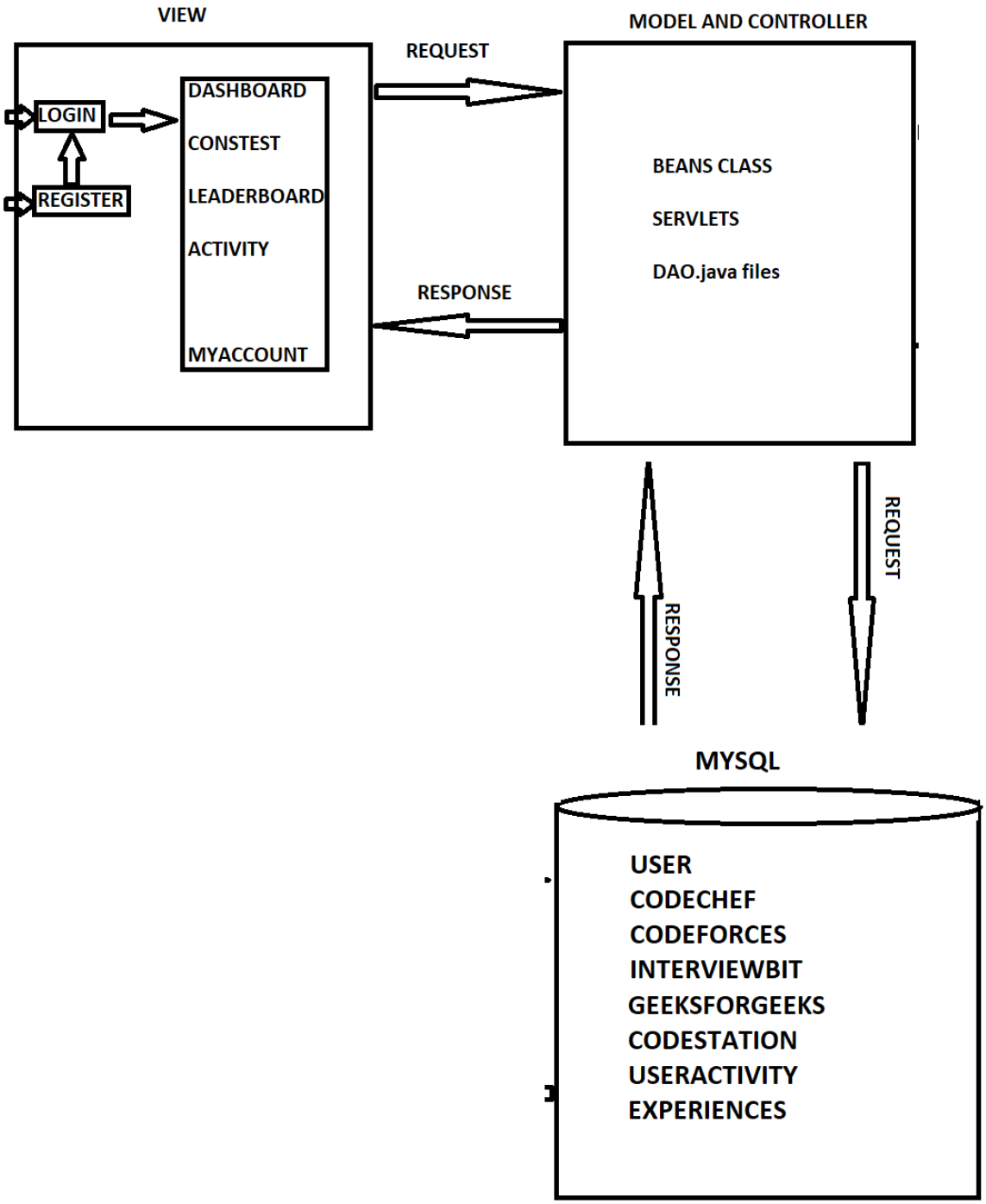
5. SYSTEM METHODOLOGY



ARCHITECTURE


Controller
(Filter/Servlet)
Database
View
(JSP)
Model
(Java Bean)
Container

WORKFLOW



6. OVERVIEW OF TECHNOLOGIES

FRONTEND	HTML 5, CSS, BOOTSTRAP 4, JAVASCRIPT, JSP
BACKEND	JAVA 11
DATABASE	MYSQL 8
SERVER	TOMCAT 9
API	JSOUP
IDE	ECLIPSE

1. HTML 5**:** The Hypertext Markup Language which is used for creating web pages and applications. Hypertext means text within the text. A text has a link within it, which is a hypertext. The markup language is a computer language that is used to apply layout and formatting conventions to a text document.

2. CSS**:** Cascade stylesheets describe how HTML elements appear on screen, paper, or other media. It saves a lot of time and allows you to control the layout of multiple web pages at the same time. This is a simple design language that aims to simplify the process of creating a website.

3. BOOTSTRAP**:** Bootstrap is an HTML, CSS, and JS library that focuses on facilitating the development of useful websites (rather than web apps). This is a front-end framework used for easier and faster web development. The main purpose of adding to a web project is to apply Bootstrap's selected colors, sizes, fonts, and layouts to that project. You can also use the JAVASCRIPT plug-in to easily create responsive designs.

4. SERVLETS AND JSP: Servlets are used to dynamically create data from users' create forms and websites and present the results. JSP and Java Server Pages are Servlet-like technologies used to build web applications. A JSP page from HTML code, embedded in one of the Java codes. The server side is requested after the client's Java code has

performed such processing, and the generated HTML page is returned to the client browser.

5. MYSQL**:** MySQL is a relational database management system based on SQL structured query language. Like other relational databases, MySQL stores data in row and column tables. Applications are used for a wide spectrum of purposes, including data warehouses, electronic commerce, and logging applications. However, MY SQL's most common use is to store data in the database.

6. JSOUP**:** JSOUP is an open-source Java library used primarily for extracting data from HTML. This is an HTML parser that can also edit and output HTML. It has a stable development line, excellent documentation, and a fluent and flexible API. You can also use JSOUP to parse and create XML.

7. TOMCAT**:** This is an open-source Java Servlet container. The main usage goal is to implement various Java Enterprise specifications such as website APIs, Java Server Pages, and Java Servlets. Developed and maintained by the open developer community with the support of the Apache Software Foundation, it is released under Apache License 2.0.

7. IMPLEMENTATION

7.1. CODING

7.2. TESTING

LOGIN PAGE:

- The user has to log in using his/her mail id and password.

```
<form method="post" action="<%= request.getContextPath() %>/UserLoginServlet">
    <h1 style="letter-spacing: 5px;font-size: 50px;color: black;font-family:Garamond, serif;padding-left: 17px;">CODESTATION <i class="fa fa-cu
    <div class="card">
      <div class="container">
        <h1 >LOGIN</h1><br>
        <table style="padding-bottom: 5px;">
            <tr><td><label>Email </label></td></tr>
            <tr><td><input type="text" name="email"></td></tr>
        </table>
        <table>
            <tr><td><label>Password </label></td></tr>
            <tr><td><input type="password" name="pass"></td></tr>
        </table>
        <br><br>
        <input type="submit" class="button" name="login" value="LOGIN" style="margin-bottom: 10px;"><br>
        <a href="Register.jsp" id="signup">Don't have account yet?</a>
      </div><br><br>
    </div>
</form>
```

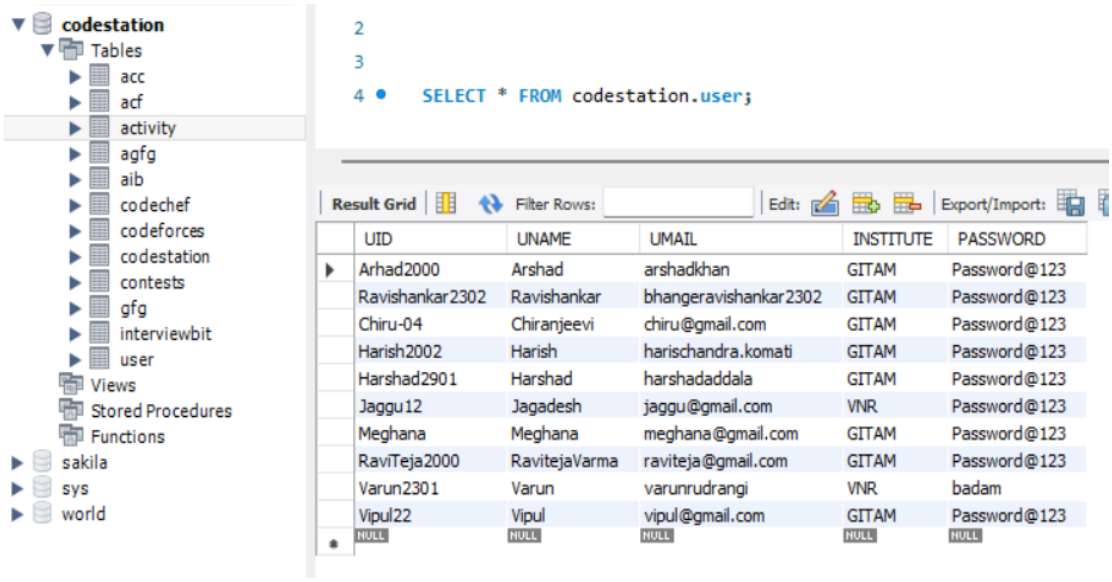
REGISTER PAGE:

- The user has to give his details like Name, Mail Id, User Id, Password, Institute for creating an account on this website.

```
<form method="post" action="<%= request.getContextPath() %>/UserServlet">
    <h1 style="letter-spacing: 5px;font-size: 50px;color: black;font-family:Garamond, serif;padding-left: 17px;">CODESTATION <i
    <div class="card">
      <div class="container">
        <h1>REGISTER</h1><br>
        <table style="padding-bottom: 5px;">
          <tr>
            <td><label>Name </label></td>
            <td><input type="text" name="name"></td>
          </tr>
          <tr>
            <td><label>Email </label></td>
            <td><input type="text" name="email"></td>
          </tr>
          <tr>
            <td><label>Create UserID </label></td>
            <td><input type="text" name="userid"></td>
          </tr>
          <tr>
            <td><label>Password </label></td>
            <td><input type="text" name="pass"></td>
          </tr>
          <tr>
            <td><label>Confirm Password </label></td>
            <td><input type="text" name="cpass"></td>
          </tr>
          <tr>
            <td><label>Institute Name </label></td>
            <td><input type="text" name="insti"></td>
          </tr>
        </table><br><br>
        <input type="submit" class="button" name="" value="REGISTER" style="margin-bottom: 10px;"><br>
        <a href="login.jsp" id="signup">Already have account?</a>
      </div><br><br>
    </div>

</form>
```

USERS TABLE:



MY ACCOUNT PAGE:

- To create an account on this website users should have an account on websites like CODECHEF, CODEFORCES, GEEKSFORGEEKS, and INTERVIEWBIT.
- After giving all user ids of all platforms, data will be fetched and stored in the database, and by default, activity will be considered Inactive.

```
<div class="column" style="float: left;width: 75%;padding: 0 10px;">
  <div class="card" style="height: 550px;"><br><br><br>
      <form method="post" action="<%= request.getContextPath() %>/PlatformServlet">
      <div class="block">
          <img src="ccimg.png" style="width:33px;height: 30px;float:left;">
          <label style="font-size: 20px;">CodeChef</label>
          <input type="text" name="cc" value="" style="width:77%;height:30px;" required>
      </div><br><br><br>
      <div class="block">
          <img src="gfgimg.png" style="width:30px;height: 26px;float:left;">
          <label style="font-size: 20px;">GeeksforGeeks</label>
          <input type="text" name="gfg" value="" style="width:77%;height:30px;" required>
      </div><br><br><br>
      <div class="block">
          <img src="cfimg.png" style="width:30px;height: 30px;float:left;">
          <label style="font-size: 20px;">CodeForces</label>
          <input type="text" name="cf" value="" style="width:77%;height:30px;" required>
      </div><br><br><br>
      <div class="block">
          <img src="ibimg.png" style="width:30px;height: 30px;float:left;">
          <label style="font-size: 20px;">InterviewBit</label>
          <input type="text" name="ib" value="" style="width:77%;height:30px;" required>
      </div><br><br><br><br><br>

      <button type="submit" class="button button3" style="background-color: #5c9e6a;">Update</button>
      </form>


  </div>
```

DASHBOARD:

- On this page users can be able to view their different platform scores, ratings as well as overall scores.
- Mathematical Equation for calculating the overall score of a user is given below

  Overall Score = (CCPS*10 +(CCR-1300) ^2/30) +(CFPS*10 +(CFR-1200) ^2/30)
  +(IBS/3) + (GFG*10)

  Where,

  CCPS = CODECHEF problems solved

  CCR = CODECHEF rating

  CFPS = CODEFORCES problems solved

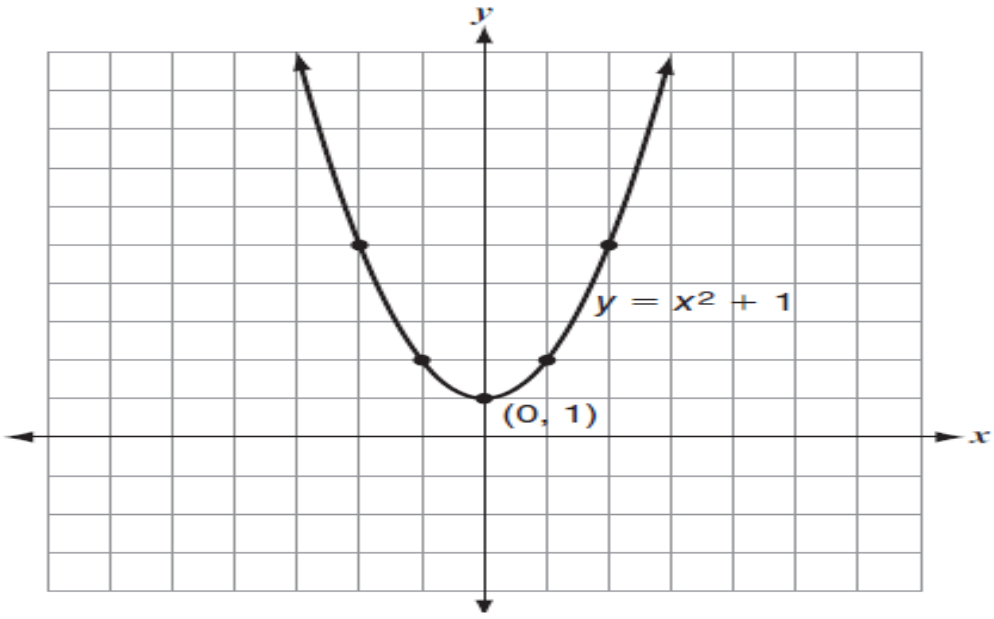
  CFR = CODEFORCES rating

  IBS = INTERVIEWBIT score

  GFG = GEEKSFORGEEKS solved problems

- For example, Codechef Score = CCPS*10 +(CCR-1300) ^2/30)

  That means the equation is in the form of y = x^2 + c where y stands for CodeChef score and x stands for rating and c stands for problems solved in the platform. Here the rate of increase in score mostly depended on the contest rating of a platform. The score increases drastically as the rating increases similar to a quadratic equation graph shown below



```
//-------------------------------------------------CODESTATION-----------------------------------------

String csid="",csint="";
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/codestation","root","meghana12");
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("SELECT UID,INSTITUTE FROM USER WHERE UMAIL='"+umail+"' ");

    while(rs.next()){
        csid = rs.getString("UID");
        csint = rs.getString("INSTITUTE");
    }
}catch(ClassNotFoundException cfe){
    cfe.printStackTrace();
}catch(SQLException cfe) {
    cfe.printStackTrace();
}
int temp=cur_rat;
int temp2=ccrating;
if(ccrating<1300){
    temp2=1300;
}
if(cur_rat<1200){
    temp=1200;
}
int csps = ccfsp + sovprob + gfgps + ibscore/300;
int ccscore=(ccfsp * 10) +( (temp2-1300)*(temp2-1300)/30  );
int cfscore=(sovprob * 10) +( (temp-1200)*(temp-1200)/30);
int interviewbitscore = ibscore/3;
int gfgsc = gfgps *10;
int csos = ((ccscore)+( cfscore  ) + interviewbitscore + (gfgsc) ) ;

System.out.println(ccscore +" || "+cfscore + " || "+interviewbitscore+" || "+gfgsc);
```

Web scraping of user's CODECHEF data:

```
String ccs = "https://www.codechef.com/users/"+ccid+"";
System.out.println(ccs);
Document doc = Jsoup.connect(ccs).get();

int ccrating,ccfsp=0,ccpsp=0,ccgr=0,cccr=0;

Elements e = doc.select("div.rating-number");
ccrating = Integer.valueOf(e.html());


Elements ee = doc.select("section.rating-data-section.problems-solved div.content h5");
int j=0;
for(Element a: ee) {
    if(j++==0) {
        String str = String.valueOf(a.html());
        ccfsp = Integer.valueOf(str.substring(14,str.length()-1));
    }
    else {
        String str = String.valueOf(a.html());
        ccpsp = Integer.valueOf(str.substring(18,str.length()-1));
    }
}

Elements eee = doc.select("div.rating-ranks ul li a strong");
int i=0;
for(Element a: eee) {
    if(i++==0) {
        String str = String.valueOf(a.html());
        if(!str.equals("Inactive"))
            ccgr = Integer.valueOf(a.html());
    }
    else {
        String str = String.valueOf(a.html());
        if(!str.equals("Inactive"))
            cccr = Integer.valueOf(a.html());
    }
}
```

```
3 •    SELECT * FROM codestation.codechef;
```

Result Grid | Filter Rows: | Edit: | Expo

CODECHEFID	UMAIL	RATING	FULL_SOV	PAR_SOV
arshad_khan63	arshadkhan	1640	10	0
chiranjeevi_19	chiru@gmail.com	1680	63	2
harish_komati	harischandra.komati	1641	46	1
harshad_2904	harshadaddala	1632	41	1
jagadish_27	jaggu@gmail.com	1475	3	0
maggi_12	meghana@gmail.com	0	0	0
raviteja_varma	raviteja@gmail.com	1620	46	0
shankar_143	bhangeravishankar2302	1655	122	6
varunrudrangi	varunrudrangi	1562	68	1
vipul_agrawal	vipul@gmail.com	1809	203	8
NULL	NULL	NULL	NULL	NULL

Web scraping of user's INTERVIEWBIT data:

```
String iburl = "https://www.interviewbit.com/profile/"+ibid;
Document ibcontest = Jsoup.connect(iburl).get();
Elements iba = ibcontest.select("div.stat.pull-left div.txt");

int ibrank=0,ibscore=0,ibstreak=0;
i=0;
for(Element ibe: iba) {
    if(i==0) {
        ibrank = Integer.valueOf(ibe.html());
    }
    else if(i==1) {
        ibscore = Integer.valueOf(ibe.html());
    }else if(i==2) {
        String str = String.valueOf(ibe.html());
        ibstreak = Integer.valueOf(str.substring(0,1));
    }
    System.out.println("Interviewbit "+ibe.html());
    i++;
}
```

```
3 •    SELECT * FROM codestation.interviewbit;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

INTERVIEWBITID	UMAIL	STREAK	SCORE	URANK
121810311004	chiru@gmail.com	0	2571	116995
abdul-arshad-khan	arshadkhan	0	2168	125897
addala-harshad	harshadaddala	0	0	499443
Bhange223	bhangeravishankar2302	0	17011	39663
chekuri-ravi-teja-varma	raviteja@gmail.com	0	2577	116882
jagadish_27	jaggu@gmail.com	0	0	497590
leelaharischandra-komati	harischandra.komati	0	1336	154584
madarapu-meghana_854	meghana@gmail.com	0	0	497578
varunrudrangi	varunrudrangi	0	28141	23245
vipul-agrawal_949	vipul@gmail.com	0	222	331264
NULL	NULL	NULL	NULL	NULL

Web scraping of user's CODEFORCES data:

```
String cfurl ="https://codeforces.com/profile/"+cfid;
System.out.println(cfid);
Document cfdoc = Jsoup.connect(cfurl).get();
//Elements a = cfdoc.select("h1.long_handle a");
Elements b = cfdoc.select("div._UserActivityFrame_counterValue");
Elements c = cfdoc.select("span.user-gray");

int sovprob=0,cur_rat=0,hig_rat=0;

i=0;
for(Element cfe: b) {
    if(i==0) {
        String str = String.valueOf(cfe.html());
        System.out.println(str);
        sovprob = Integer.valueOf(str.substring(0,str.length()-9));
        System.out.println("Sov prob");
    }
    else if(i==7) {
        cur_rat = Integer.valueOf(cfe.html());
    }else if(i==9) {
        hig_rat = Integer.valueOf(cfe.html());
    }
    i++;
}
System.out.println("Ratings");
i=0;
for(Element cfe: c) {
    if(i==1) {
        cur_rat = Integer.valueOf(cfe.html());
    }else if(i==3) {
        hig_rat = Integer.valueOf(cfe.html());
    }
    i++;
}
```

```
3 •    SELECT * FROM codestation.codeforces;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

CODEFORCES	UMAIL	CUR_RATING	HIG_RATING	SOV_PROB
	harshadaddala	0	0	0
ABDUL_ARSHAD_KHAN	arshadkhan	953	967	33
Chiroo19	chiru@gmail.com	1130	1130	47
harish_komati	harischandra.komati	984	984	38
Jagadish_27	jaggu@gmail.com	0	0	12
maggi12meghana	meghana@gmail.com	0	0	0
Ravishankarbhange2302	bhangeravishankar2302	957	1041	57
raviteja__varma	raviteja@gmail.com	438	438	36
Sholey	harshadaddala	748	748	40
varunrudrangi	varunrudrangi	0	0	62
VipulAgrawal31	vipul@gmail.com	0	0	163
NULL	NULL	NULL	NULL	NULL

Web scraping of user's GEEKSFORGEEKS data:

```
String gfgs = "https://auth.geeksforgeeks.org/user/"+gfgid+"/practice/";

System.out.println(gfgid);
Document contest = Jsoup.connect(gfgs).get();
Elements gfga = contest.select("div.mdl-grid div.mdl-cell.mdl-cell--6-col.mdl-cell--12-col-phone.textBold");
Elements gfgb = contest.select("div.mdl-cell.mdl-cell--6-col.mdl-cell--12-col-phone a");
Elements gfgc = contest.select("div.mdl-cell.mdl-cell--4-col.mdl-cell--12-col-phone.textBold span");
j=0;
int gfgcs=15,gfgps=0,gfgms=0;
//System.out.println(a.html());
for(Element gfge: gfga) {
    if(j==0) {
        String str = String.valueOf(gfge.html());
        gfgcs = Integer.valueOf(str.substring(34,str.length()));
        System.out.println("Coding score : "+gfgcs);
    }
    j++;
}
if(gfgcs!=0) {
    String gfgss = String.valueOf(gfgb.html());
    System.out.println(gfgb.html());
    gfgps = Integer.valueOf(gfgss.substring(22,gfgss.length()));
    System.out.println("Problems solved: "+gfgps);
}
j=0;
for(Element gfge: gfgc) {
    if(j++==0) {
        String str = String.valueOf(gfge.html());
        gfgms = Integer.valueOf(str);
        System.out.println("Montly Streak: "+gfgms);
    }
}
```

```
5 •   SELECT * FROM codestation.gfg;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wra

GFGID	UMAIL	SOLVED_PROB	CODING_SCOR	MON_STREAK
121810311004	chiru@gmail.com	85	159	0
arshadkhan63	arshadkhan	79	194	0
bhangeravishankar2302	bhangeravishankar2302	219	413	0
harischandrakomati	harischandra.komati	57	109	0
harshadips12	harshadaddala	3	2	0
mmaggi12345	meghana@gmail.com	0	0	0
nerellajagadish	jaggu@gmail.com	0	0	0
ravitejavarma2000	raviteja@gmail.com	84	164	0
varunrudrangi	varunrudrangi	6	21	0
vipulagrawal31	vipul@gmail.com	235	411	0
NULL	NULL	NULL	NULL	NULL

LEADERBOARD:

- We have two leaderboards (Global leaderboard and Leaderboard by Institute)

```
<%
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/codestation","root","meghana12");
        Statement st = con.createStatement();
String sql=
("SELECT U.UNAME, U.UMAIL, U.INSTITUTE, CS.OVERALL_SCORE FROM CODESTATION AS CS, USER AS U WHERE U.UMAIL=CS.UMAIL ORDER BY CS.OVERALL_SCORE DESC ;");
        ResultSet rs = st.executeQuery(sql);
        int i=1;
        while(rs.next()) {
            String name = rs.getString("UNAME");
            String email = rs.getString("UMAIL");
            String score = rs.getString("OVERALL_SCORE");
            String institute = rs.getString("INSTITUTE");
            %>
              <tr>
               <td><%out.println(i++); %></td>
               <td><%=name %></td>
               <td><%=email %></td>
               <td><%=score %></td>
               <td><%=institute %></td>
              </tr>
            <%
        }
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
%>
```

```
4 •    SELECT * FROM codestation.codestation;
```

Result Grid | Filter Rows: | Export:

	UMAIL	SOL_PROB	OVERALL_SCORE
▶	bhangeravishankar2302	454	13850
	harischandra.komati	145	5731
	harshadaddala	84	4514
	arshadkhan	129	5795
	varunrudrangi	229	13028
	raviteja@gmail.com	174	5932
	meghana@gmail.com	0	0
	jaggu@gmail.com	15	1170
	chiru@gmail.com	203	7620
	vipul@gmail.com	601	14720

ACTIVITY PAGE:

- On this page users can view their weekly activity and daily activity from 4 platforms
- In this application, we created a thread called SiteMapThread which runs in the background and performs some background tasks like updating users' activity. The user activity table will be updated every 24 hours

```
482
483 public class SiteMapThread implements Runnable {
484
485     private ServletContext context;
486
487     public SiteMapThread(ServletContext context) {
488         this.context = context;
489     }
490
491     @Override
492     public void run() {
493         do {
494             try {
495                 Call c = new Call();
496                 int day=1;
497                 c.userCall(day);
498                 Thread.sleep(1000*60*60*24);
499                 day++;
500             } catch (InterruptedException e) {
501                 e.printStackTrace();
502             }
503         }while(true);
504     }
505
506 }
```

- In daily activity, data will be updated by storing current date data in day7 and previous date data will be stored in previous-> previous Data. Just like adding a node in LinkedList and deleting the first node in LinkedList.
- Along with these dates' update, if any scores or ratings changed in users' coding profiles they will be updated in the database to track weekly activity.

```
int ccsun=0,ccmon=0,cctus=0,ccwed=0,ccthu=0,ccfri=0,ccsat=0;
int cfsun=0,cfmon=0,cftus=0,cfwed=0,cfthu=0,cffri=0,cfsat=0;
int gfgsun=0,gfgmon=0,gfgtus=0,gfgwed=0,gfgthu=0,gfgfri=0,gfgsat=0;
int ibsun=0,ibmon=0,ibtus=0,ibwed=0,ibthu=0,ibfri=0,ibsat=0;
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/codestation","root","meghana12");
    Statement st = con.createStatement();
    String sql = ("SELECT SUN, MON, TUS, WED, THU, FRI, SAT FROM ACC WHERE UMAIL='"+umail+"' ;");
    ResultSet rs = st.executeQuery(sql);
    while(rs.next()) {
          ccsun = Integer.parseInt(rs.getString("SUN"));
          ccmon = Integer.parseInt(rs.getString("MON"));
          cctus = Integer.parseInt(rs.getString("TUS"));
          ccwed = Integer.parseInt(rs.getString("WED"));
          ccthu = Integer.parseInt(rs.getString("THU"));
          ccfri = Integer.parseInt(rs.getString("FRI"));
          ccsat = Integer.parseInt(rs.getString("SAT"));

    }


    sql = ("SELECT SUN, MON, TUS, WED, THU, FRI, SAT FROM ACF WHERE UMAIL='"+umail+"' ;");
    rs = st.executeQuery(sql);
    while(rs.next()) {
          cfsun = Integer.parseInt(rs.getString("SUN"));
          cfmon = Integer.parseInt(rs.getString("MON"));
          cftus = Integer.parseInt(rs.getString("TUS"));
          cfwed = Integer.parseInt(rs.getString("WED"));
          cfthu = Integer.parseInt(rs.getString("THU"));
          cffri = Integer.parseInt(rs.getString("FRI"));
          cfsat = Integer.parseInt(rs.getString("SAT"));

    }


    sql = ("SELECT SUN, MON, TUS, WED, THU, FRI, SAT FROM AGFG WHERE UMAIL='"+umail+"' ;");
```

```
<canvas id="myChart" style="width:100%;max-width:600px"></canvas><br>
<p>Codechef - BROWN | CodeForces - RED | GeeksforGeeks - GREEN | INTERVIEWBIT - BLUE</p>
<script>
var xValues = ['Day1','Day2','Day3','Day4','Day5','Day6','Day7'];

new Chart("myChart", {
  type: "line",
  data: {
    labels: xValues,
    datasets: [{
      data: [<%=ccsun%>,<%=ccmon%>,<%=cctus%>,<%=ccwed%>,<%=ccthu%>,<%=ccfri%>,<%=ccsat%>],
      borderColor: "brown",
      fill: false
    }, {
      data: [<%=cfsun%>,<%=cfmon%>,<%=cftus%>,<%=cfwed%>,<%=cfthu%>,<%=cffri%>,<%=cfsat%>],
      borderColor: "red",
      fill: false
    }, {
      data: [<%=gfgsun%>,<%=gfgmon%>,<%=gfgtus%>,<%=gfgwed%>,<%=gfgthu%>,<%=gfgfri%>,<%=gfgsat%>],
      borderColor: "green",
      fill: false
    },{
      data: [<%=ibsun%>,<%=ibmon%>,<%=ibtus%>,<%=ibwed%>,<%=ibthu%>,<%=ibfri%>,<%=ibsat%>],
      borderColor: "blue",
      fill: false
    }]
  },
  options: {
    legend: {display: false}
  }
});
</script>

</center>
```

```
<tr>
    <td>CodeChef</td>
    <%
        if( (lastw.compareTo(ccd))<=0){
    %>
            <td style="color:green;font-weight: bold;">Active</td>
    <%
        }else{
    %>
            <td style="color:Red">Inactive</td>
    <%
        }
    %>
</tr>

<tr>
    <td>CodeForces</td>
    <%

        if( (lastw.compareTo(cfd))<=0){
    %>
            <td style="color:green;font-weight: bold;">Active</td>
    <%
        }else{
    %>
            <td style="color:Red">Inactive</td>
```

```
LocalDate today = LocalDate.now();
LocalDate lastw = today.plusDays(-7);
//System.out.println("Last date is "+lastw);
LocalDate ccd = null;
LocalDate cfd = null;
LocalDate hed = null;
//System.out.println("Here is the maill "+umail);
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/codestation","root","meghana12");
    Statement st = con.createStatement();
    ResultSet rs = st.executeQuery("SELECT CC,CF FROM ACTIVITY WHERE UMAIL='"+umail+"' ");

    while(rs.next()){
        ccd = LocalDate.parse((rs.getString("CC")));
        //System.out.println("Last date is "+lastw+"---"+ccd);
        cfd = LocalDate.parse((rs.getString("CF")));
        //System.out.println("Last date is "+lastw+"---"+cfd);

    }
}catch (ClassNotFoundException cfe) {
    cfe.printStackTrace();
}catch (SQLException cfe) {
    cfe.printStackTrace();
}
```

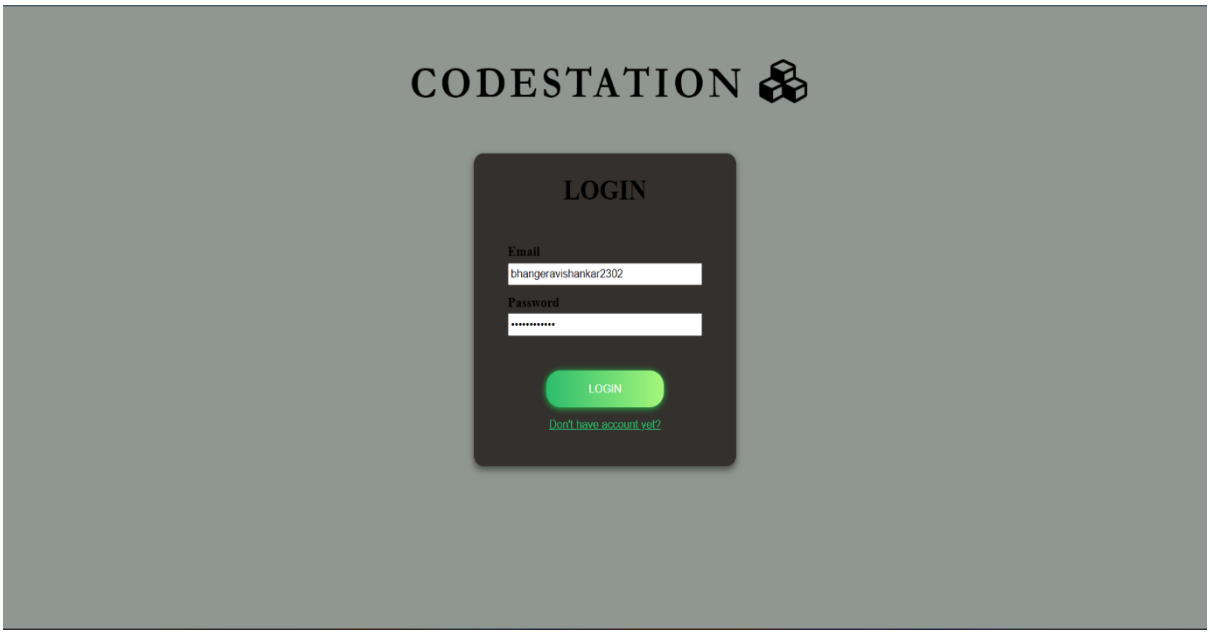
```
4 •  SELECT * FROM codestation.activity;
5
```

Result Grid | Filter Rows: | Export:

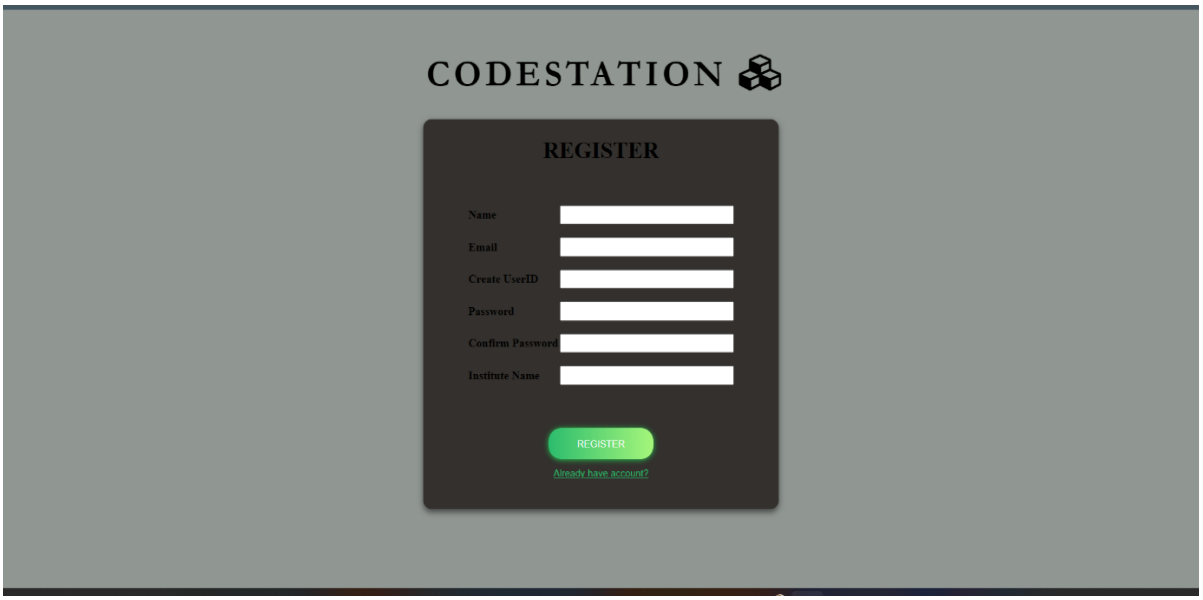
UMAIL	CC	CF
bhangeravishankar2302	2022-03-25	2022-03-30
arshadkhan	2022-03-28	2022-03-28
harischandra.komati	2022-03-01	2022-03-01
harshadaddala	2022-03-01	2022-03-01
varunrudrangi	2022-03-01	2022-03-01
jaggu@gmail.com	2022-03-01	2022-03-01
meghana@gmail.com	2022-03-01	2022-03-01
raviteja@gmail.com	2022-03-01	2022-03-01
vipul@gmail.com	2022-03-01	2022-03-01
chiru@gmail.com	2022-03-01	2022-03-01

8. RESULTS AND DISCUSSIONS

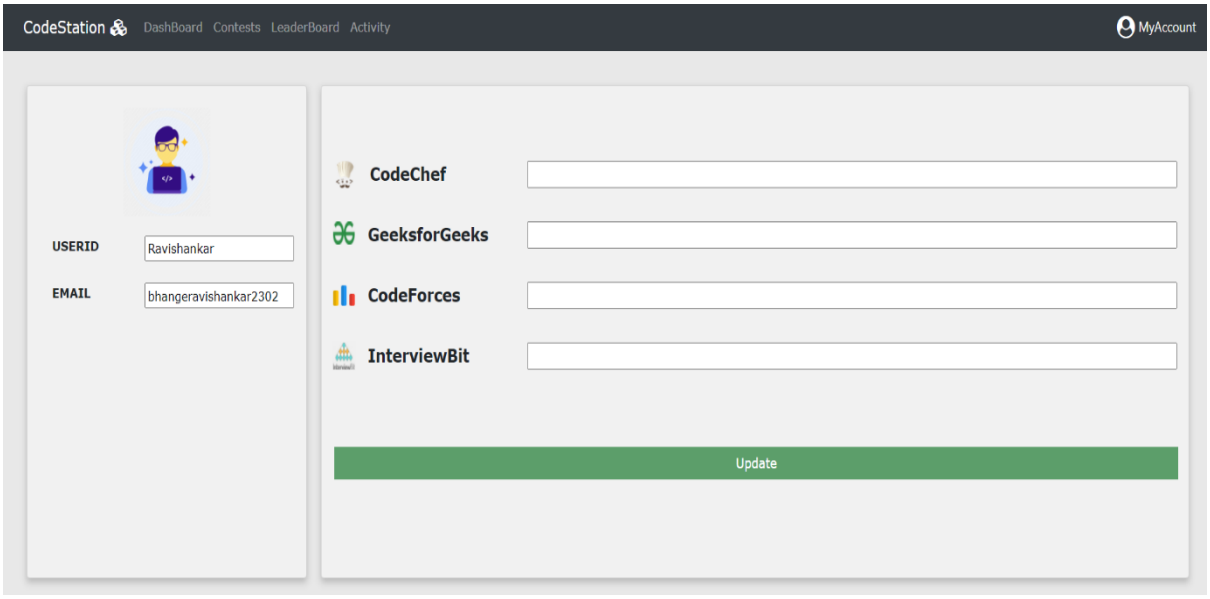
LOGIN PAGE:



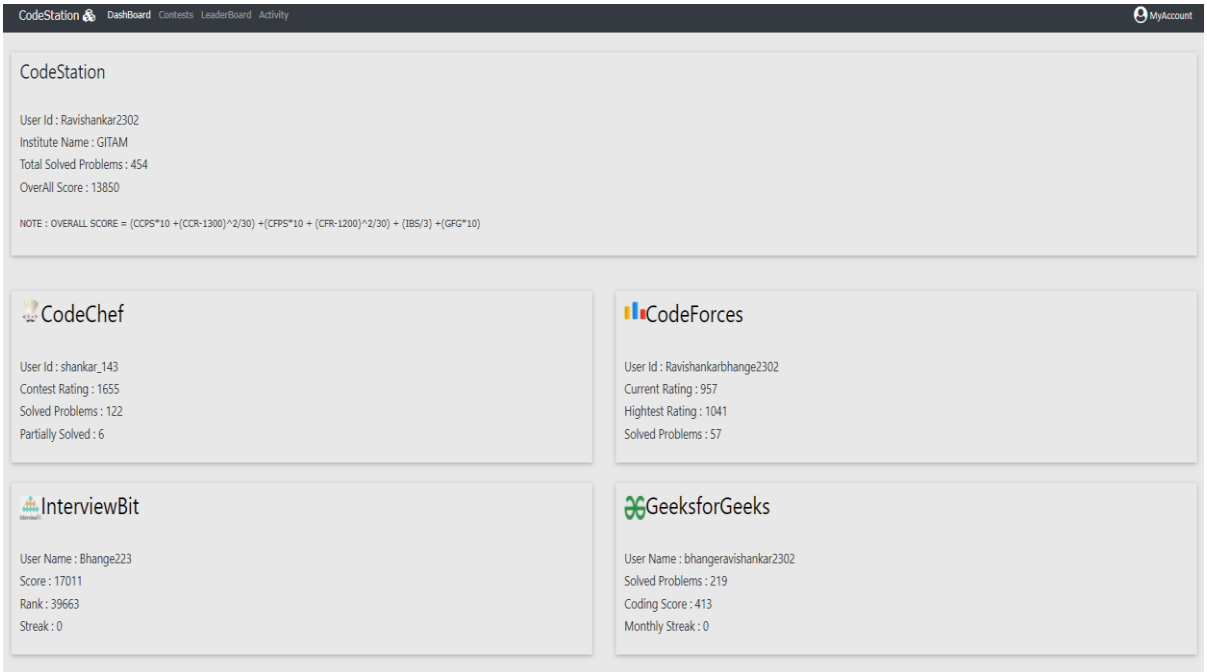
REGISTER PAGE:



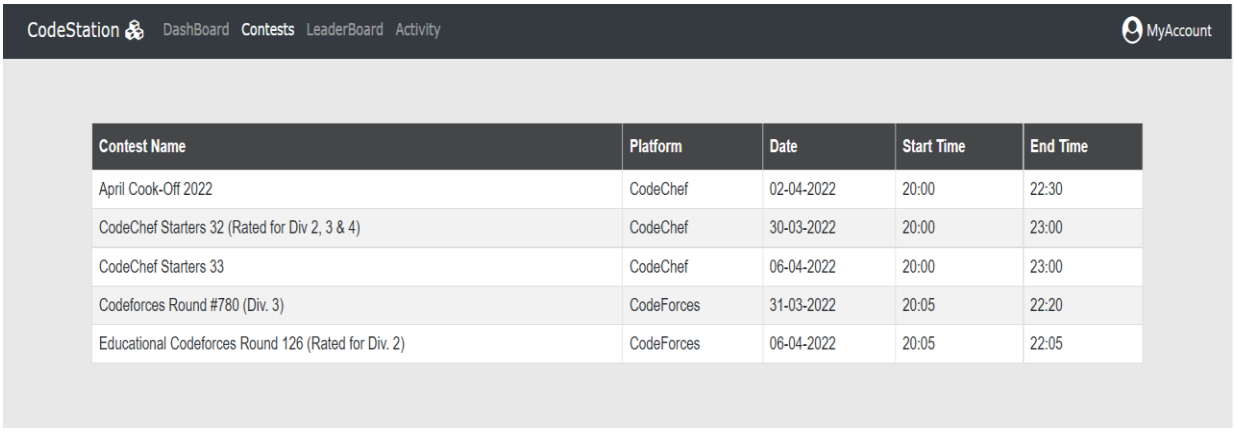
MYACCOUNT PAGE:



DASHBOARD PAGE:



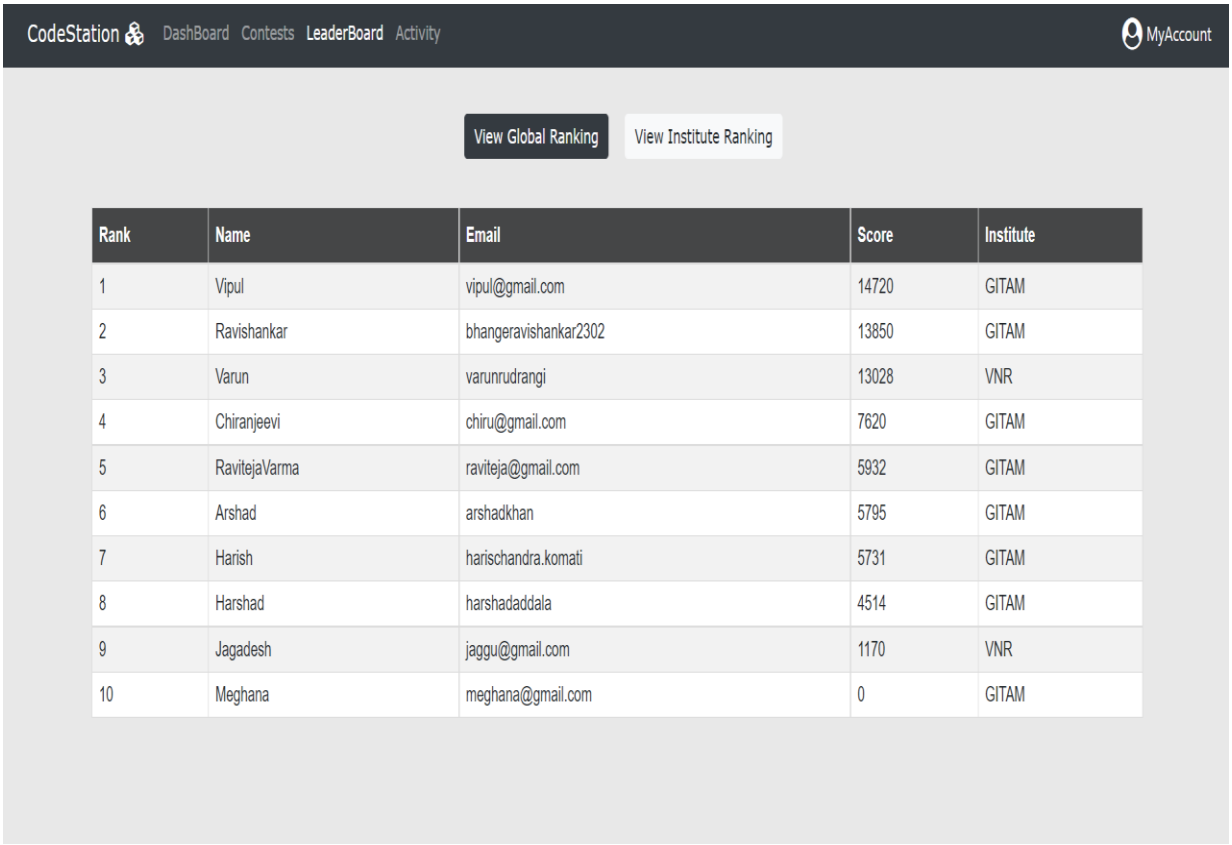
CONTEST PAGE:



Contest Name	Platform	Date	Start Time	End Time
April Cook-Off 2022	CodeChef	02-04-2022	20:00	22:30
CodeChef Starters 32 (Rated for Div 2, 3 & 4)	CodeChef	30-03-2022	20:00	23:00
CodeChef Starters 33	CodeChef	06-04-2022	20:00	23:00
Codeforces Round #780 (Div. 3)	CodeForces	31-03-2022	20:05	22:20
Educational Codeforces Round 126 (Rated for Div. 2)	CodeForces	06-04-2022	20:05	22:05

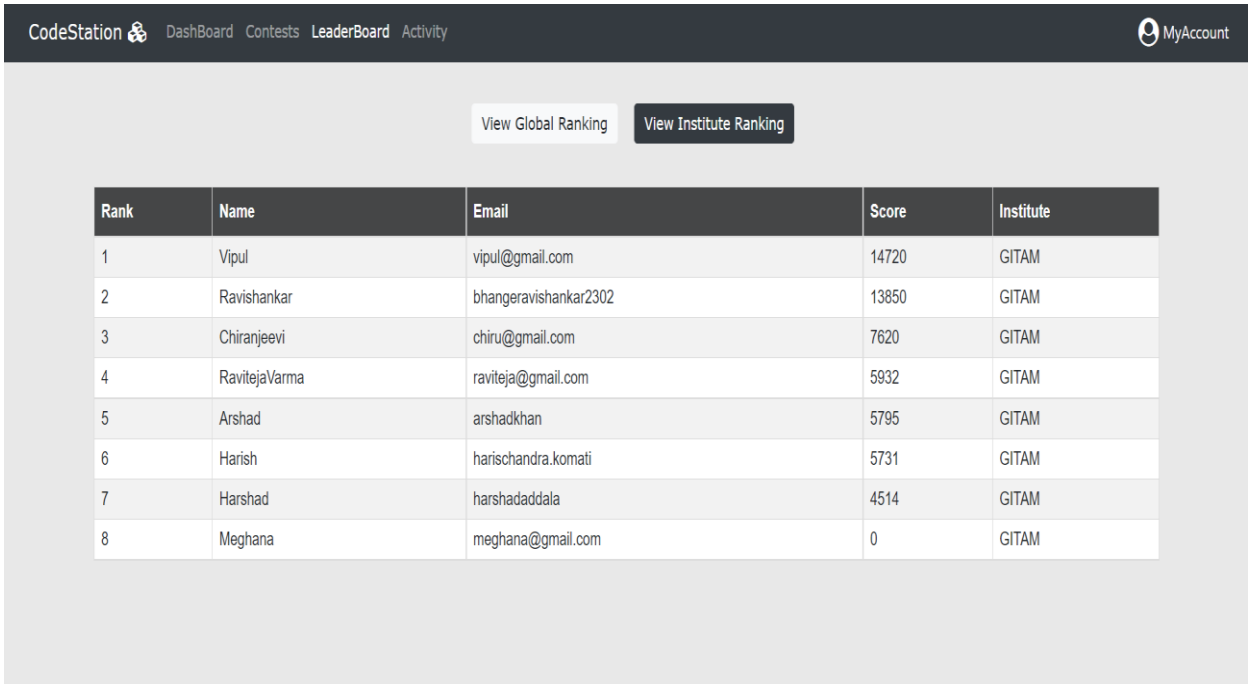
LEADERBOARD PAGE:

Global Ranking:



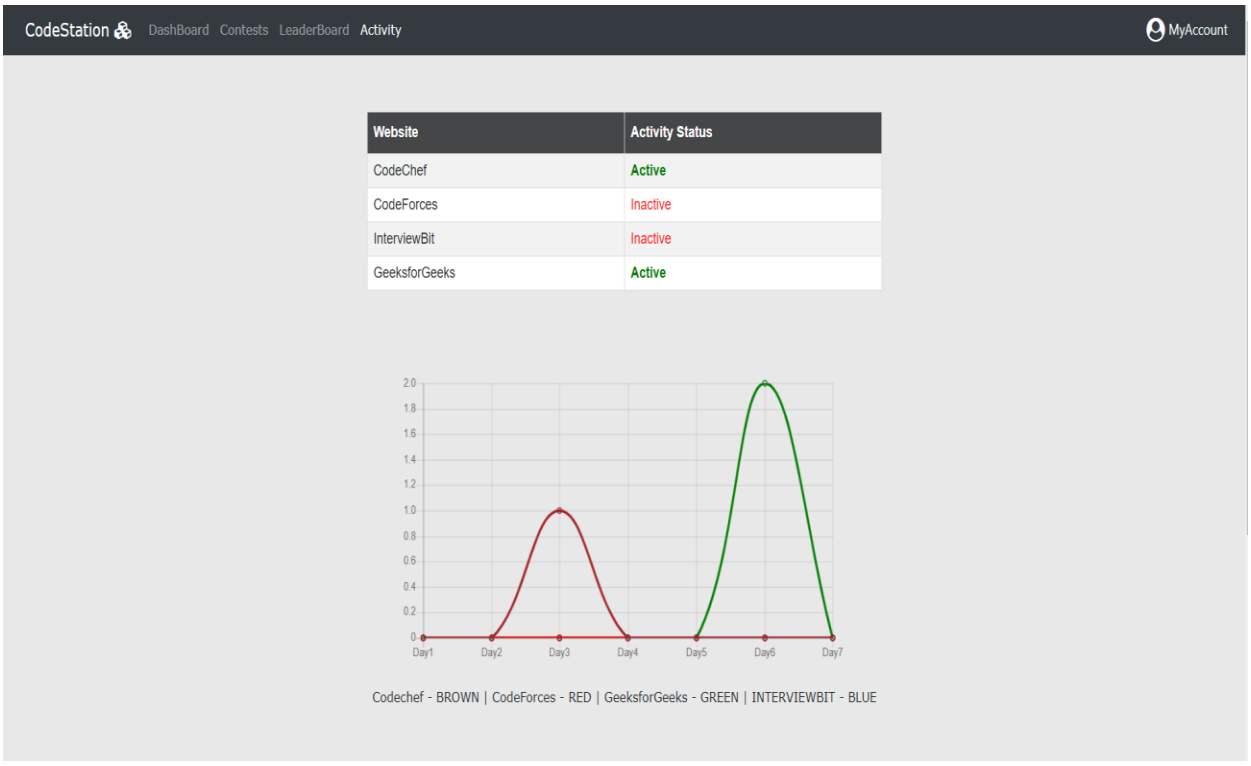
Rank	Name	Email	Score	Institute
1	Vipul	vipul@gmail.com	14720	GITAM
2	Ravishankar	bhangeravishankar2302	13850	GITAM
3	Varun	varunrudrangi	13028	VNR
4	Chiranjeevi	chiru@gmail.com	7620	GITAM
5	RavitejaVarma	raviteja@gmail.com	5932	GITAM
6	Arshad	arshadkhan	5795	GITAM
7	Harish	harischandra.komati	5731	GITAM
8	Harshad	harshadaddala	4514	GITAM
9	Jagadesh	jaggu@gmail.com	1170	VNR
10	Meghana	meghana@gmail.com	0	GITAM

Institute Ranking:



Rank	Name	Email	Score	Institute
1	Vipul	vipul@gmail.com	14720	GITAM
2	Ravishankar	bhangeravishankar2302	13850	GITAM
3	Chiranjeevi	chiru@gmail.com	7620	GITAM
4	RavitejaVarma	raviteja@gmail.com	5932	GITAM
5	Arshad	arshadkhan	5795	GITAM
6	Harish	harischandra.komati	5731	GITAM
7	Harshad	harshadaddala	4514	GITAM
8	Meghana	meghana@gmail.com	0	GITAM

ACTIVITY PAGE:



Codeforces Activity Table:

```
4 •    SELECT * FROM codestation.acf;
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

UMAIL	SUN	MON	TUS	WED	THU	FRI	SAT
jaggu@gmail.com	0	0	0	0	0	0	0
arshadkhan	0	1	0	0	0	0	0
harischandra.komati	0	0	0	0	0	0	0
harshadaddala	0	0	0	0	0	0	0
meghana@gmail.com	0	0	0	0	0	0	0
raviteja@gmail.com	0	0	0	0	0	0	0
varunrudrangi	0	0	0	0	0	0	0
chiru@gmail.com	0	0	0	0	0	0	0
vipul@gmail.com	0	0	0	0	0	0	0
bhangeravishankar2302	0	0	1	0	0	0	0

GeeksforGeeks Activity Table:

```
1 •    SELECT * FROM codestation.agfg;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

UMAIL	SUN	MON	TUS	WED	THU	FRI	SAT
bhangeravishankar2302	0	0	0	0	0	2	0
jaggu@gmail.com	0	0	0	0	0	0	0
arshadkhan	0	0	0	0	0	0	0
harischandra.komati	0	0	0	0	0	0	0
harshadaddala	0	0	0	0	0	0	0
meghana@gmail.com	0	0	0	0	0	0	0
raviteja@gmail.com	0	0	0	0	0	0	0
varunrudrangi	0	0	0	0	0	0	0
chiru@gmail.com	0	0	0	0	0	0	0
vipul@gmail.com	0	0	0	0	0	0	0

CodeChef Activity Table:

```
4 •    SELECT * FROM codestation.acc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

UMAIL	SUN	MON	TUS	WED	THU	FRI	SAT
bhangeravishankar2302	0	0	1	0	0	0	0
jaggu@gmail.com	0	0	0	0	0	0	0
arshadkhan	0	1	0	0	0	0	0
harischandra.komati	0	0	0	0	0	0	0
harshadaddala	0	0	0	0	0	0	0
meghana@gmail.com	0	0	0	0	0	0	0
raviteja@gmail.com	0	0	0	0	0	0	0
varunrudrangi	0	0	0	0	0	0	0
chiru@gmail.com	0	0	0	0	0	0	0
vipul@gmail.com	0	0	0	0	0	0	0

InterviewBit Activity Table:

```
3 •    SELECT * FROM codestation.aib;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

UMAIL	SUN	MON	TUS	WED	THU	FRI	SAT
bhangeravishankar2302	0	0	0	0	0	0	0
jaggu@gmail.com	0	0	0	0	0	0	0
arshadkhan	0	0	0	0	0	0	0
harischandra.komati	0	0	0	0	0	0	0
harshadaddala	0	0	0	0	0	0	0
meghana@gmail.com	0	0	0	0	0	0	0
raviteja@gmail.com	0	0	0	0	0	0	0
varunrudrangi	0	0	0	0	0	0	0
chiru@gmail.com	0	0	0	0	0	0	0
vipul@gmail.com	0	0	0	0	0	0	0

```
4 •  SELECT * FROM codestation.activity;
5
```

Result Grid | Filter Rows: | Export:

UMAIL	CC	CF
bhangeravishankar2302	2022-03-25	2022-03-30
arshadkhan	2022-03-28	2022-03-28
harischandra.komati	2022-03-01	2022-03-01
harshadaddala	2022-03-01	2022-03-01
varunrudrangi	2022-03-01	2022-03-01
jaggu@gmail.com	2022-03-01	2022-03-01
meghana@gmail.com	2022-03-01	2022-03-01
raviteja@gmail.com	2022-03-01	2022-03-01
vipul@gmail.com	2022-03-01	2022-03-01
chiru@gmail.com	2022-03-01	2022-03-01

9. CONCLUSION AND FUTURE SCOPE

As we have gone through different modules in the project we can say that this website will be very helpful for students who are willing to improve their skills sets in competitive programming. In this website, we have seen four platforms with ratings and scores of a user. Activity will be tracked every 24 hours. This website can be made more graphical interface by adding graphs and some other way of representation. This project can be extended by increasing the number of platforms and adding some more features like topic recommendations based on the type of problems solved in each platform. If this idea has been implemented then it will be very helpful to check on which topics they should concentrate and they can see how many problems they have solved on each topic. This can be implemented by problem tags in which each problem will be tagged with some topics. We can also add features like follow/following so that they can follow their friends to check their profiles easily. With this, we can also add one more leaderboard among friends for each user so that they can actively check their position among their friends. It would be great if add user submissions on this website so that users can view all submissions in one place and view problems according to tags, max success rate, favorite, etc. so that we will be redirected to that particular website to solve the problem. This website might become a hub if we implement all the above ideas. By adding all these features it will be helpful to students which makes them very easy to track their profiles and all. And even if they want to show their coding profiles they can just share their codestation profile which contains all scores from different platforms and having an overall score from that will create a portfolio

10. REFERENCES

- https://www.w3.org/standards/webdesign/htmlcss
- https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)
- https://www.upgrad.com/blog/jsp-vs-servlet/
- https://www.123-reg.co.uk/support/servers/what-is-mysql-and-why-do-i-need-it/
- https://www.baeldung.com/java-with-jsoup
- https://n6host.com/blog/what-is-tomcat-6-reasons-you-should-use-tomcat/