International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

CODING ASSESSMENT PORTAL

A.Bhavishya[1], A.Neelima Devi[2], B.Bhavana[3],
M.Swapna[4]
Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana, India

Abstract . The 'Coding Assessment Portal' is a tool for students to improve their coding skills. Becoming a programmer isn't just about learning the syntax and the concepts of a programming language, it's about figuring out how to use that knowledge to make programs.

This portal consists of programming challenges in which a description about the problem statement is given, the user can choose any desirable programming language and write the code in an editor. The code written should be compiled successfully and should produce correct output. Based on the student's performance, the admin will assign a score.

In order to build this portal we need to work on various sub-modules which include: Registration and Login page for users and admins, Dashboard for user panel and admin panel, Integrating Code editor and Compiler in User panel. In the user panel once the student login's he'll be able to find Dashboard which contains tabs like view challenges, Python Assessment, Java Assessment and Logout. In Order to write assessment the students should click on view challenges where a list of Programming challenges are displayed, then they should click on the solve challenge button and write the code. If a student wants to check his/her code for errors they can use a tab named java assessment or python assessment which has a code editor, an in-built compiler and an output screen. This also includes security features like single sign-in i.e disabling multiple logins using a single account, disabling the users to record the test or taking screenshots during the test.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Keywords: Coding Assessment, programming language, Registration page, Login page, User panel, Admin panel, Python Assessment, Java Assessment, Compiler, Code editor, Sign-in.

1. Introduction

1.1 About Project

This portal consists of programming challenges in which a description about the problem statement is given, the user can choose any desirable programming language and write the code in an editor. The code written should be compiled successfully and should produce correct output. Based on the student's performance, the admin will assign a score.

The online test created has following stages

- Login
- Test
- Result

LOGIN:-

There is a quality login window because this is more secure than other login forms as in a normal login window there are multiple logins available so that more than one person can access to test with their individual login. But in this project there is only one login id i.e. administrator id and password by which a person enters the site. Hence it is more secure and reliable than previously used on-line test simulators.

TEST:

Test page is the most creative and important page in this project. It consists of 2 modules namely:

- Subject selection:

From the given choices the candidate can select his field (like JAVA and python etc) for taking on with the test.

- Utilities: It includes

❖ Skip and come back to the question afterwards if needed.

❖ Gives the list of attempted and unattempted questions and can go to any question directly and can either attempt or change the answer of the already attempted question.

1.2 Objectives of the Project:

- To test the knowledge or learning of a candidate.
- To select suitable candidates from a huge pool of applicants.
- To identify the strengths and weaknesses of the test taker
- To efficiently evaluate the candidate thoroughly through a fully automated system that not only saves a lot of time but also gives fast results.

1.3 SCOPE:

Scope of this project is very broad in terms of other manually taking exams. Few of them are:-

- This can be used in educational institutions as well as in the corporate world.
- Can be used anywhere any time as it is a web based application(user Location doesn't matter).
- No restriction that the examiner has to be present when the candidate takes the test.

1.4 Advantages:

- Secure
- Easy to use
- Reliable and accurate
- No need of examiner

1.5 Disadvantages:

- The current system is very time consuming.
- It is very difficult to analyse the exam manually.
- To take the exam of more candidates more invigilators are required but no need of invigilator in case of online exam.
- Results are not precise as calculation and evaluations are done manually.
- The chances of paper leakage are more in the current system as compared to the proposed system.

1.6 Applications :

- In Software Employee Hiring Process
- In Colleges
- In Training Institutes

1.7 Hardware and Software Requirements:

Hardware Requirements:

- Processor: 1.5GHz or above
- RAM: 4GB or more
- HDD: 100GB or above

Software Requirements:

- Operating System: Windows 10 or higher
- Front end: HTML, CSS, JavaScript
- Back end: Python
- Server: Django Framework
- Database: Mysql
- Querying Language: Sql

2. Literature Survey

2.1 Existing System:

- The whole process of assigning tests and evaluating their scores after the test, was done manually till date.
- Processing the test paper i.e. checking and distributing respective scores used to take time when the software was not installed.
- Evaluation of a candidate's coding skills was done through traditional pen and paper methods.
- Execution of the code is done in their own IDE's and then the source code file is sent to their respective evaluator's or examiners.

DISADVANTAGES OF CURRENT SYSTEM:

- The current system is very time consuming.
- It is very difficult to analyse the exam manually.
- To take the exam of more candidates more invigilators are required but no need of invigilator in case of online exam.
- Results are not precise as calculation and evaluations are done manually.
- The chances of paper leakage are more in the current system as compared to the proposed system.
- Result processing takes more time as it is done manually.

2.2 Proposed System:

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

How does our proposed system overcome these problems ?

- Students can write the test from any remote location.
- An IDE consisting of problem statement, problem description, choice of selecting desirable programming language, editor and output screen .
- Admin can evaluate the student's code once the student submits.

  Students can view his score once admin assigns score after evaluation.
- Students can use the in-built compilers for practising.

To implement this project we have designed following modules

1) New User Signup: using this module users can sign up with the application.

Fig 2.1 New User Signup

2) User Login: using this module, users can login to the application.

Fig 2.2:User Login

3) Python Assessment: using this module a user can write a python program and then the application will compile and run code and based on correctness the application will give a grade.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

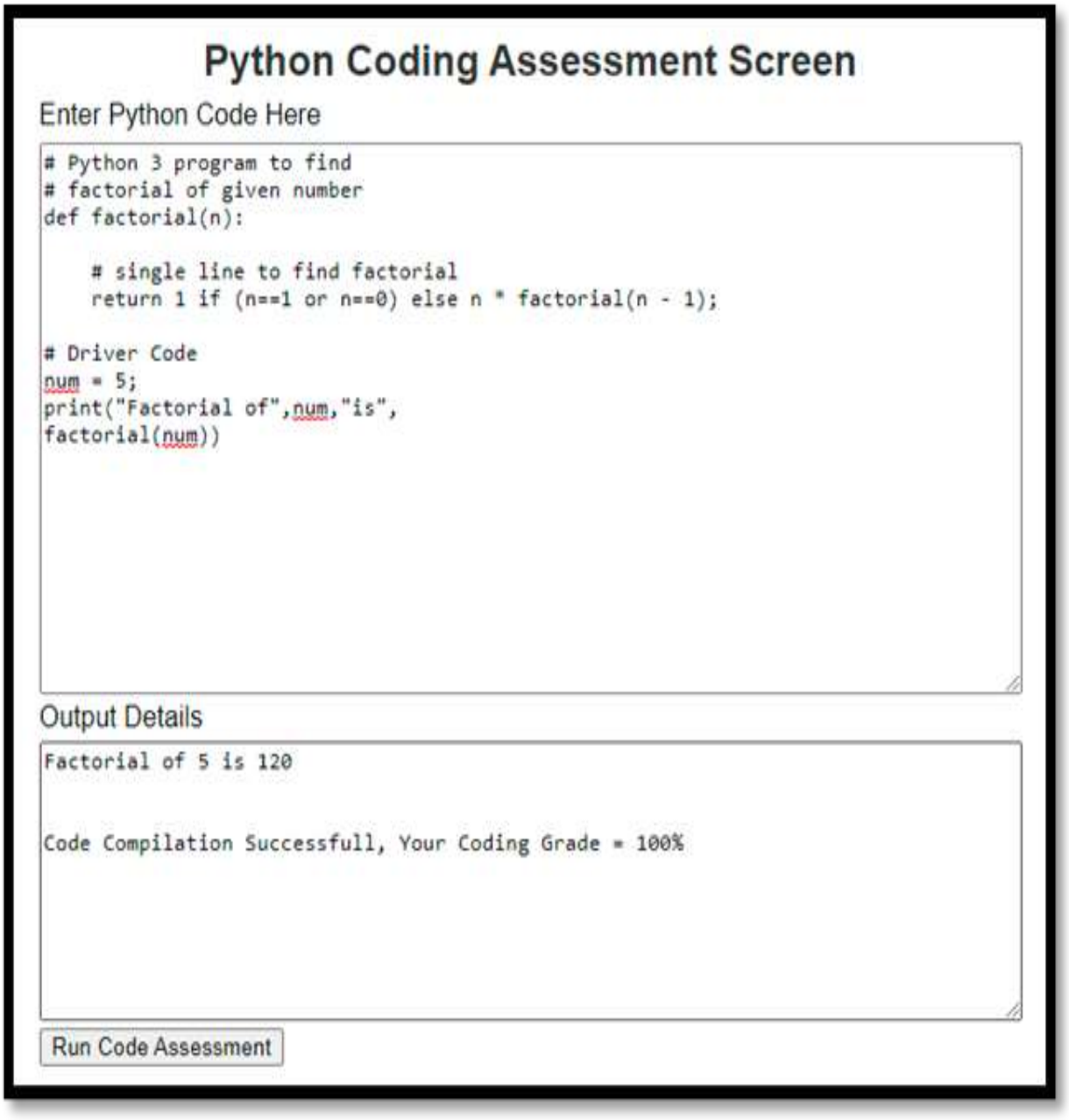

Fig 2.3:Python coding Assessment Screen

4) Java Assessment: using this module, users can write a java program and then the application will compile and run code and based on correctness the application will give a grade.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

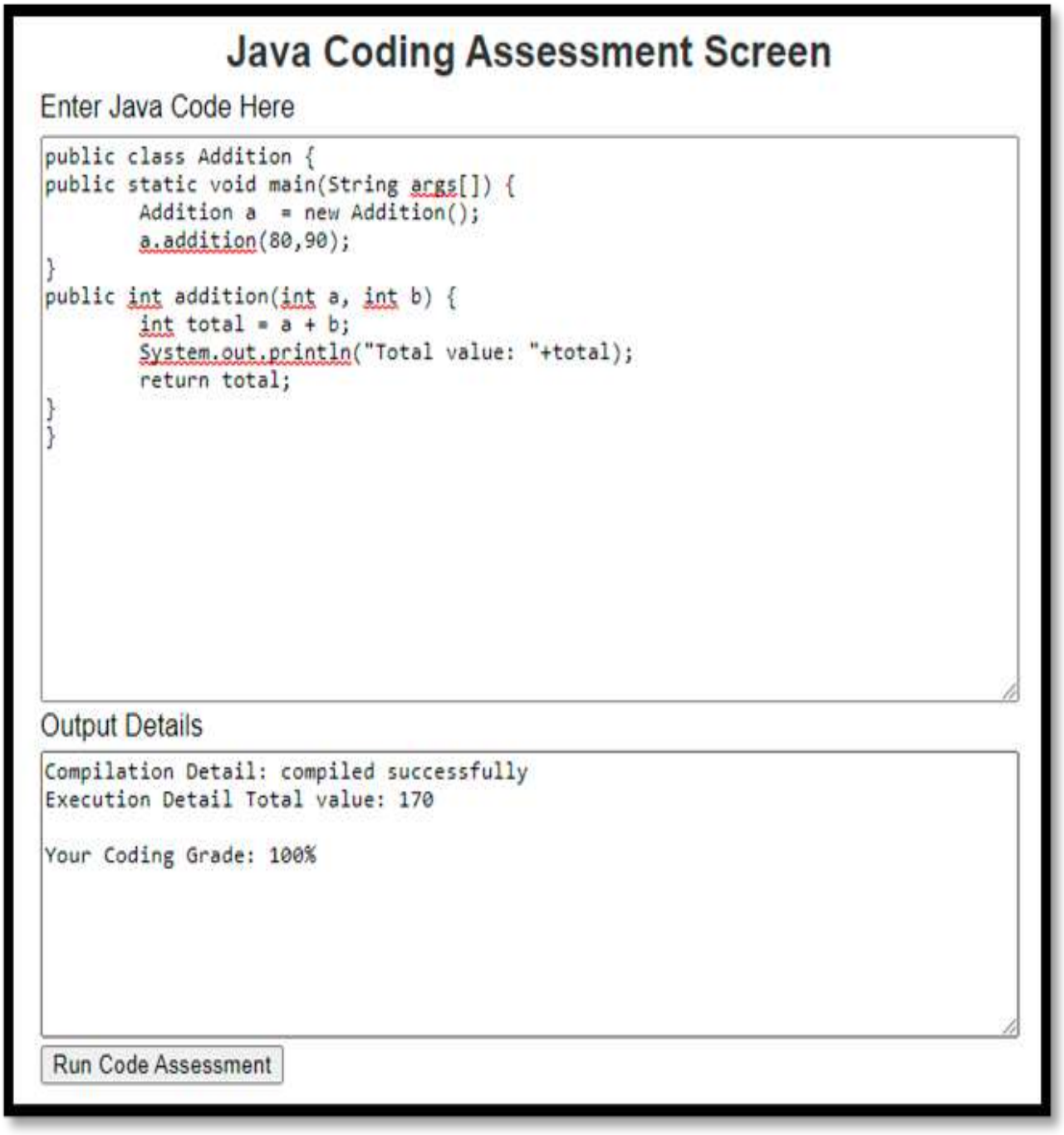

Fig 2.4:Java Coding Assessment Screen

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

5) Admin Login; admin can login to the application using username and password as ‘admin’ and ‘admin’ and after login admin can view all registered users.

Fig 2.5:Admin Login Screen

3.Proposed Architecture:

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

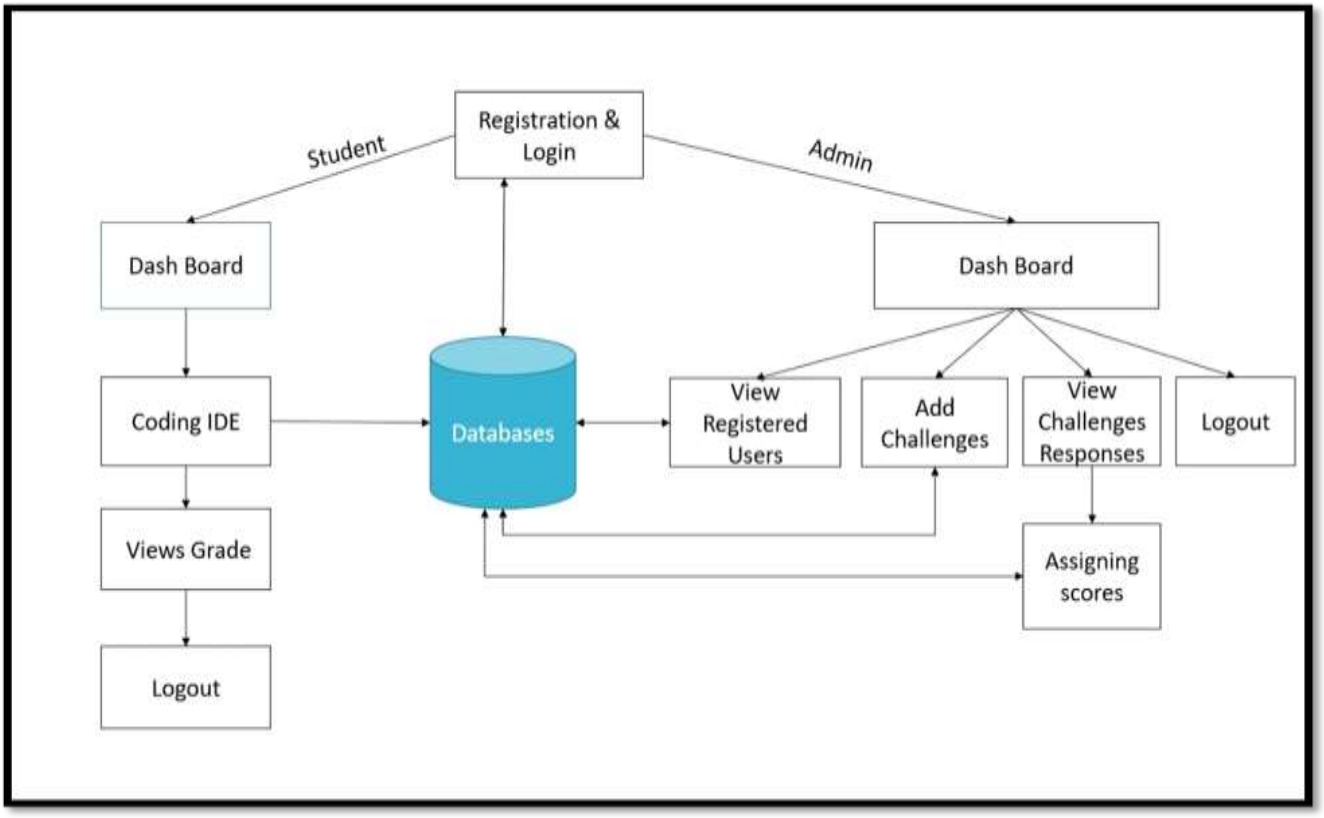

Fig 3.1: Flow diagram of coding assessment portal

The main components of the coding assessment portal are:

1.Registration page for users and login pages for both admin and users

2.Dashboard for students

3.Dashboard for admin

4. Coding IDE for students

1.Registration page:

Registration only happens the first time you access the system. It is a way to check your credentials. Every time after your initial registration, you will log on to the system using the username and password you created.

The registration page is only for the users and not for the admin. Admin can directly login with his credentials.

The registration page has following fields:

- User name
- Password
- Contact number
- Email id
- Address

After entering all the details you should click on submit, then you can move to the login page and you can login every time directly by entering username and password.

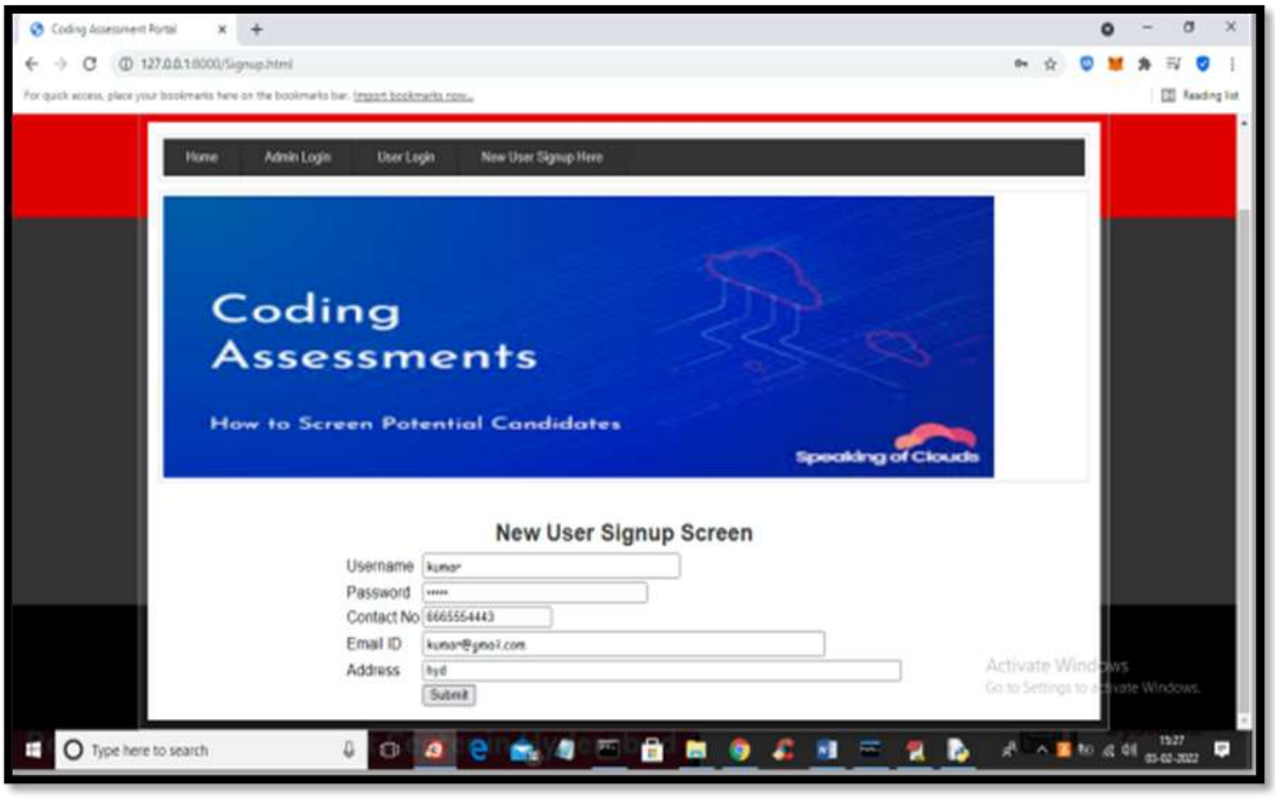

Fig 3.2:Registration Page

2. Login page for users:

A login page is a web page or an entry page to a website that requires user identification and authentication, regularly performed by entering a username and password combination. Logins may provide access to an entire site or part of a website. Logging in not only provides site access for the user, but also allows the website to track user actions and behaviour.

There is a quality login window because this is more secure than other login forms as in a normal login window there are multiple logins available so that more than one person can access to test with their individual login. But in this project there is only one login id i.e. administrator id and password by which a person enters the site. Hence it is more secure and reliable than previously used on-line test simulators.

There are two login pages, one for users and one for admin.

Fig 3.3: User login page

Fig 3.4: Admin login page

3. Dashboard for students:

As soon as they login, a dashboard appears which has different options like

- View challenges
- Java assessment
- Python assessment

In java assessment tab and python assessment tab we have a code editor and an inbuilt compiler with output screen. The candidate can check for errors in his code here.

The candidates can write their code in two languages which are python and java.

4.Dashboard for admin:

The admin dashboard has few functionalities such as:

- View registered users
- View responses
- Add challenges

The admin can add the programming challenges by writing the problem description.

The admin can then view the responses after the candidate writes his code and submits.

The admin can then evaluate the code and assign him a score.

5.Coding IDE:

The portal has inbuilt coding IDEs in which you can write your code in a code editor and then compile. And the output will be displayed on the screen. The candidate can check for errors and debug and then go back to the challenge page and write the code and submit. Once he writes the code and submits, the admin can view the code and assign the score accordingly.

4.Implementation

4.1 Algorithm:

VIEWS

1.Import all the necessary libraries.

2.Define views / methods that are implemented in our model so that whenever a user requests for that functionality that particular method will be executed.

3.Whenever user requests for Java Assessment or Python Assessment, application should take him to that respective page.

4.Similarly if user requests for index, login, signup application should take him to the respective pages.

5.If admin requests to login, add challenges then application should take him to the respective page.

6.Methods to be defined to compile java code where the user can know the number of errors, where the error has occurred, or the code is compiled and executed successfully.

7.Method to define functionality of Accept Challenge where user can accept the challenge given by admin.

8.Method to get Score where user can view his score after Admin assigns score to that student.

9.Method to View admin challenges through which the user can know whether he has already answered the challenge or not. That is, if he answers the challenge score will be assigned to him else it shows click here to accept if the challenge is not yet accepted.

10. Method to View Users where admin can view number of registered users.

11.Method for User Login Action where the necessary credentials of the user are collected from the signup process.

12.Method for Sign Up action where user has to give the necessary details like username, email, password, contact number and address.

13. Method to Add new challenge where different challenge id's will be generated for different challenges.

14.Method to accept a challenge where if code is successfully executed the user can submit and get a popup that your answer is accepted.

15.Method to view a challenge if the score is not yet assigned.

16.Method to get Answer so that admin can receive the answer submitted by the user.

17.Method to assign a score to the challenge.

4.2 Code Implementation :

 # Import all the necessary libraries

```
from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

import os

import pymysql

import subprocess

from django.http import HttpResponse

from datetime import date


global uname
```

 # Views definition

```
def JavaAssessment(request):
        if request.method == 'GET':
        return render(request, 'JavaAssessment.html', {})

def PythonAssessment(request):
        if request.method == 'GET':
        return render(request, 'PythonAssessment.html', {})

def index(request):
        if request.method == 'GET':
        return render(request, 'index.html', {})

def AdminLogin(request):
        if request.method == 'GET':
        return render(request, 'AdminLogin.html', {})

def UserLogin(request):
```

```
    if request.method == 'GET':
    return render(request, 'UserLogin.html', {})


def Signup(request):
    if request.method == 'GET':
    return render(request, 'Signup.html', {})


def AddChallenges(request):
    if request.method == 'GET':
    return render(request, 'AddChallenges.html', {})
```

 # Code to compile Java code

```
def CompileJava(request):
    if request.method == 'GET':
    code = request.GET.get('mytext', False)
    code = code.replace("@","\n")
    codes = code.split("\n")
    cname = ""
    for i in range(len(codes)):
```

```
temp = codes[i].split(" ")
if len(temp) >= 2:
        if temp[0].strip() == 'class' or temp[1].strip() == 'class':
        if temp[0].strip() == 'class':
        cname = temp[1].strip()
        if temp[1].strip() == 'class':
        cname = temp[2].strip()
print(cname)
f = open(cname+".java", "w")
f.write(code)
f.close()
calljava = subprocess.Popen(["javac",cname+".java"], stderr=subprocess.PIPE)
compiled = calljava.stderr.read()
compile_errors = compiled.decode().strip()
length = len(compile_errors)
print("length : "+str(length))
output = ""
if length == 0:
output += "Compilation Detail: compiled successfully\n"
calljava = subprocess.Popen(["java",cname, "10","20"], stdout=subprocess.PIPE)
```

```
compiled = calljava.stdout.read()

output += "Execution Detail "+compiled.decode()+"\nYour Coding Grade: 100%"

else:

arr = compile_errors.split("\n")

errors = arr[len(arr)-1]

err = errors.split(" ")

err = int(err[0])

grade = err / len(codes)

output += compile_errors+"\n\nYour Coding Grade: "+str(grade)

if os.path.exists(cname+".java"):

os.remove(cname+".java")

if os.path.exists(cname+".class"):

os.remove(cname+".class")

return HttpResponse(output, content_type="text/plain")
```

 # Code to compile Python code

```
def CompilePython(request):

    if request.method == 'GET':

    code = request.GET.get('mytext', False)
```

```
code = code.replace("@","\n")

codes = code.split("\n")

print(code)

f = open("test.py", "w")

f.write(code)

f.close()

callpython = subprocess.Popen(["python","test.py", "40","20"], stderr=subprocess.PIPE)

error = callpython.stderr.read()

error = error.decode()

output = error

if len(error) == 0:

callpython = subprocess.Popen(["python","test.py", "40","20"], stdout=subprocess.PIPE)

output = callpython.stdout.read()

output = output.decode()+"\n\nCode Compilation Successfull, Your Coding Grade = 100%"

#print("Execution Detail "+output)

else:

arr = len(error.split("\n")) / 2

grade = len(codes) / arr

output = "Errors occured: "+error+"\n\nYour Coding grade: "+str(grade)

return HttpResponse(output, content_type="text/plain")
```

 # View for Admin Login Action

```
def AdminLoginAction(request):
        global uname
        if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if username == 'admin' and password == 'admin':
        uname = username
        context= {'data':'welcome '+uname}
        return render(request, 'AdminScreen.html', context)
        else:
        context= {'data':'invalid login details'}
        return render(request, 'AdminLogin.html', context)
```

 # View to Accept Challenge

```
def AcceptChallenge(request):
```

```
    if request.method == 'GET':
        global uname
        cid = request.GET.get('t1', False)
        output = '<td><input type="text" name="t1" style="font-family: Comic Sans MS" size="30" value="'+str(cid)+'" readonly></td></tr>'
        context= {'data1':output}
        return render(request, 'AcceptChallenge.html', context)
```

 # View to get Score

```
def getScore(sname,cid):
    score = 'none'
    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
    with con:
    cur = con.cursor()
    cur.execute("select * from challenge_result")
    rows = cur.fetchall()
    for row in rows:
    if row[0] == cid and row[1] == sname:
```

```
            score = row[3]
            break
    return score
```

View to view the admin challenges

```
def ViewAdminChallenge(request):
    if request.method == 'GET':
    global uname
    output = ""
    font = '<font size="" color="black">'
      con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
    with con:
    cur = con.cursor()
    cur.execute("select * from challenges")
    rows = cur.fetchall()
    for row in rows:
            output += '<tr><td>'+font+str(row[0])+'</td>'
            output += '<td>'+font+row[1]+'</td>'
```

```
            output += '<td>'+font+str(row[2])+'</td>'
            score = getScore(uname,row[0])
            if score == 'none':
            output+='<td><a href=\'AcceptChallenge?t1='+str(row[0])+'\'><font size=3 color=black>Click Here to Answer</font></a></td></tr>'
            else:
            output += '<td>'+font+'Already Submitted. Your Score: '+score+'</td>'
        context= {'data':output}
        return render(request, 'ViewAdminChallenge.html', context)
```

 # View to view the users

```
def ViewUsers(request):
        if request.method == 'GET':
    global uname
        output = ""
        font = '<font size="" color="black">'
        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
        with con:
```

```
        cur = con.cursor()
        cur.execute("select * from signup")
        rows = cur.fetchall()
        for row in rows:
            output += '<tr><td>'+font+str(row[0])+'</td>'
            output += '<td>'+font+row[1]+'</td>'
            output += '<td>'+font+row[2]+'</td>'
            output += '<td>'+font+row[3]+'</td>'
            output += '<td>'+font+row[4]+'</td>'
        context= {'data':output}
        return render(request, 'ViewUsers.html', context)
```

 # View to define User Login action

```
def UserLoginAction(request):
    global uname
    if request.method == 'POST':
    username = request.POST.get('t1', False)
    password = request.POST.get('t2', False)
    usertype = request.POST.get('t3', False)
```

```
index = 0
con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
with con:
cur = con.cursor()
cur.execute("select username,password FROM signup")
rows = cur.fetchall()
for row in rows:
    if row[0] == username and password == row[1]:
    uname = username
    index = 1
    break
page = "UserLogin.html"
if index == 1:
page = "UserScreen.html"
if page != "UserLogin.html":
context= {'data':'welcome '+uname}
return render(request, page, context)
else:
context= {'data':'invalid login details'}
```

```
    return render(request, page, context)
```

 # View to define user signup action

```
def SignupAction(request):
    if request.method == 'POST':
    username = request.POST.get('t1', False)
    password = request.POST.get('t2', False)
    contact = request.POST.get('t3', False)
    email = request.POST.get('t4', False)
    address = request.POST.get('t5', False)
    output = "none"
    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
    with con:
    cur = con.cursor()
    cur.execute("select username FROM signup")
    rows = cur.fetchall()
      for row in rows:
          if row[0] == username:
```

```
            output = username+" Username already exists"
            break
    if output == 'none':
    db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
    db_cursor = db_connection.cursor()
    student_sql_query = "INSERT INTO signup(username,password,contact_no,email,address) VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"
    db_cursor.execute(student_sql_query)
    db_connection.commit()
    print(db_cursor.rowcount, "Record Inserted")
    if db_cursor.rowcount == 1:
            output = 'Signup Process Completed'
    context= {'data':output}
    return render(request, 'Signup.html', context)
```

 # View to Add Challenges

```
def AddChallengesAction(request):
    if request.method == 'POST':
```

```
challenge = request.POST.get('t1', False)

today = date.today()

count = 0

con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select count(*) FROM challenges")

rows = cur.fetchall()

for row in rows:

        count = row[0]

count = count + 1

db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')

db_cursor = db_connection.cursor()

student_sql_query = "INSERT INTO challenges(challenge_id,challenge_data,challenge_date) VALUES("'+str(count)+'","'+challenge+'","'+str(today)+'")"

db_cursor.execute(student_sql_query)

db_connection.commit()

print(db_cursor.rowcount, "Record Inserted")
```

```
    if db_cursor.rowcount == 1:
    output = 'Challenge details added with ID: '+str(count)
    context= {'data':output}
    return render(request, 'AddChallenges.html', context)
```

View to Accept Challenge

```
def AcceptChallengeAction(request):
    if request.method == 'POST':
    global uname
    cid = request.POST.get('t1', False)
    answer = request.POST.get('t2', False)
    db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
    db_cursor = db_connection.cursor()
    student_sql_query = "INSERT INTO challenge_result(challenge_id,student_name,student_answer,score) VALUES('"+cid+"','"+uname+"','"+answer+"','Pending')"
    db_cursor.execute(student_sql_query)
      db_connection.commit()
    print(db_cursor.rowcount, "Record Inserted")
```

```
        if db_cursor.rowcount == 1:
        output = 'Your answer accepted'
        context= {'data':output}
        return render(request, 'UserScreen.html', context)
```

 # View to view challenges

```
def ViewChallenges(request):
        if request.method == 'GET':
        global uname
        output = ""
        font = '<font size="" color="black">'
        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')
        with con:
        cur = con.cursor()
        cur.execute("select * from challenge_result where score='Pending'")
        rows = cur.fetchall()
        for row in rows:
                output += '<tr><td>'+font+str(row[0])+'</td>'
```

```
        output += '<td>'+font+row[1]+'</td>'

        output += '<td>'+font+row[2]+'</td>'

        output+='<td><a     href=\'ChallengeScore?t1='+str(row[0])+'&t2='+row[1]+'\'><font size=3 color=black>Click Here to Give Score</font></a></td></tr>'

        context= {'data':output}

        return render(request, 'ViewChallenges.html', context)
```

 # View to get answer given by user

```
def getAnswer(cid,sname):

        answer = "none"

        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')

        with con:

        cur = con.cursor()

        cur.execute("select student_answer from challenge_result where challenge_id='"+str(cid)+"' and student_name='"+sname+"'")

        rows = cur.fetchall()

        for row in rows:

        answer = row[0]

        break
```

```
    return answer
```

 # Views to get challenge score and the respective action to be taken

```
def ChallengeScore(request):
    if request.method == 'GET':
    cid = request.GET.get('t1', False)
    sname = request.GET.get('t2', False)
    answer = getAnswer(cid,sname)
    output = '<td><input type="text" name="t1" style="font-family: Comic Sans MS" size="30" value="'+cid+'" readonly></td></tr>'
    output+= '<tr><td><font size="" color="black">Student Name</b></td>'
    output += '<td><input type="text" name="t2" style="font-family: Comic Sans MS" size="30" value="'+sname+'" readonly></td></tr>'
    output+= '<tr><td><font size="" color="black">Student Answer</b></td>'
    output+= '<td><textarea name="t3" rows="15" cols="80">'+answer+'</textarea></td></tr>'
    context= {'data1':output}
    return render(request, 'ChallengeScore.html', context)

def ChallengeScoreAction(request):
```

```
if request.method == 'POST':

cid = request.POST.get('t1', False)

sname = request.POST.get('t2', False)

score = request.POST.get('t4', False)

db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'Assessment',charset='utf8')

db_cursor = db_connection.cursor()

student_sql_query = "update challenge_result set score='"+score+"' where challenge_id='"+str(cid)+"' and student_name='"+sname+"'"

db_cursor.execute(student_sql_query)

db_connection.commit()

print(db_cursor.rowcount, "Record Inserted")

if db_cursor.rowcount == 1:

output = 'Score successfully assigned '

context= {'data':output}

return render(request, 'AdminScreen.html', context)
```

TEMPLATES:

Different templates used in our project are:

- AcceptChallenge
- AddChallenges
- AdminLogin
- AdminScreen
- ChallengeScore
- Index
- JavaAssessment
- PythonAssessment
- Signup
- UserLogin
- UserScreen
- ViewAdminChallenge
- ViewChallenges
- ViewUsers

ACCEPT CHALLENGE

```
{% load static %}

<html>

<head>

<title>Coding Assessment Portal</title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="{% static 'style.css' %}" rel="stylesheet" type="text/css" />

<script language="javascript">
```

```
function validate(formObj)
{
if(formObj.t2.value.length==0)
{
alert("Please Enter your answer");
formObj.t2.focus();
return false;
}
return true;
}
</script>
<script language="javascript" type="text/javascript" src="datetimepicker.js">
</script>
</head>
<body>
<div class="main">
<div class="main_resize">
<div class="header">
<div class="logo">
 <h1><span>Coding Assessment Portal</span><small></small></h1>
```

```
        </div>

        </div>

        <div class="content">

        <div class="content_bg">

        <div class="menu_nav">

        <ul>

        <li><a href="{% url 'PythonAssessment' %}">Python Assessment</a></li>

        <li><a href="{% url 'JavaAssessment' %}">Java Assessment</a></li>

        <li><a href="{% url 'ViewAdminChallenge' %}">View Challenges</a></li>

        <li><a href="{% url 'index' %}">Logout</a></li>

    </ul>

        </div>

        <div class="hbg"><img src="{% static 'images/header_images.jpg' %}" width="915" height="286" alt="" /></div>

<center>

<form name="f1" method="post" action={% url 'AcceptChallengeAction' %} onsubmit="return validate(this);">

{% csrf_token %}<br/>

   <h2><b>Add Challenges Screen</b></h2>



        <font size="" color="red"><center>{{ data }}</center></font>
```

```
<table align="center" width="30%" >
        <tr><td><font size="" color="black">Challenge ID</b></td>
                {{ data1|safe }}
        <tr><td><font size="" color="black">Your Answer</b></td>
                <td><textarea name="t2" rows="15" cols="80"></textarea></td></tr>

        <tr><td></td><td><input type="submit" value="Submit"></td>
</table>
</div>
</div>

</body>
</html>
```

JAVA ASSESSMENT

```
{% load static %}
<html>
<head>
```

```
<title>Coding Assessment Portal</title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="{% static 'style.css' %}" rel="stylesheet" type="text/css" />

<script language="javascript">

        function validate(formObj)

        {

        if(formObj.t1.value.length==0)

        {

        alert("Please Enter  java code");

        formObj.t1.focus();

        return false;

        }

        return true;

        }

        function runCode() {

        var request = new XMLHttpRequest();

        var input = document.getElementById("t1").value;

        input = input.replace(/\r\n|\r|\n/g,"@");

        input = encodeURIComponent(input)

        request.open("GET", "http://127.0.0.1:8000/CompileJava?mytext="+input);
```

```
        request.onreadystatechange = function() {
        if(this.readyState === 4 && this.status === 200) {
           data = this.responseText
                document.getElementById("t2").innerHTML = data;
        }
        };
        request.send();
}
focusInput = function()
{
        document.focus();
};


processKeyEvent = function(eventType, event)
{
        if (window.event)
        {
        event = window.event;
        }
        if(event.keyCode == 44)
```

```
        {
        alert("sorry screenchots cannot be captured");
        return(false);
        }
}
processKeyUp = function(event)
{
        processKeyEvent("onkeyup", event);
};

processKeyDown = function(event)
{
        processKeyEvent("onkeydown", event);
};

document.onkeyup = processKeyUp;
document.onkeydown = processKeyDown;
        </script>

</head>
```

```
<body oncontextmenu="return false;">

<div class="main">

  <div class="main_resize">

        <div class="header">

        <div class="logo">

        <h1><span>Coding Assessment Portal</span><small></small></h1>

        </div>

        </div>

        <div class="content">

        <div class="content_bg">

        <div class="menu_nav">

        <ul>

        <li><a href="{% url 'PythonAssessment' %}">Python Assessment</a></li>

          <li><a href="{% url 'JavaAssessment' %}">Java Assessment</a></li>

          <li><a href="{% url 'ViewAdminChallenge' %}">View Challenges</a></li>

        <li><a href="{% url 'index' %}">Logout</a></li>

   </ul>

        </div>

        <div class="hbg"><img src="{% static 'images/header_images.jpg' %}" width="915"
height="286" alt="" /></div>
```

```
                                <center>

{% csrf_token %}<br/>

  <h2><b>Java Coding Assessment Screen</b></h2>


        <font size="" color="red"><center>{{ data }}</center></font>


        <table align="center" width="30%" >

        <tr><td><font size="" color="black">Enter Java Code Here</b></td><tr>

        <tr><td><textarea name="t1" rows="20" cols="80" id="t1"></textarea></td></tr>


        <tr><td><font size="" color="black">Output Details</b></td></tr>

        <tr><td><textarea name="t2" rows="10" cols="80" id="t2"></textarea></td></tr>


        <tr><td><button type="button" onclick="runCode()">Run Code Assessment</button></td>

        </table>

        </div>

        </div>

        </body>

</html>
```

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

5.Results

1. In the below screen click on 'New User Signup Here' link to add a new user and to get the below screen.

Fig 5.1 : Coding Assessment Portal

2. In the above screen, the user is entering signup details and then pressing the 'Submit' button to get below screen.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Fig 5.2: New User Signup Screen

3.In the below screen in red colour text we can see the signup process completed

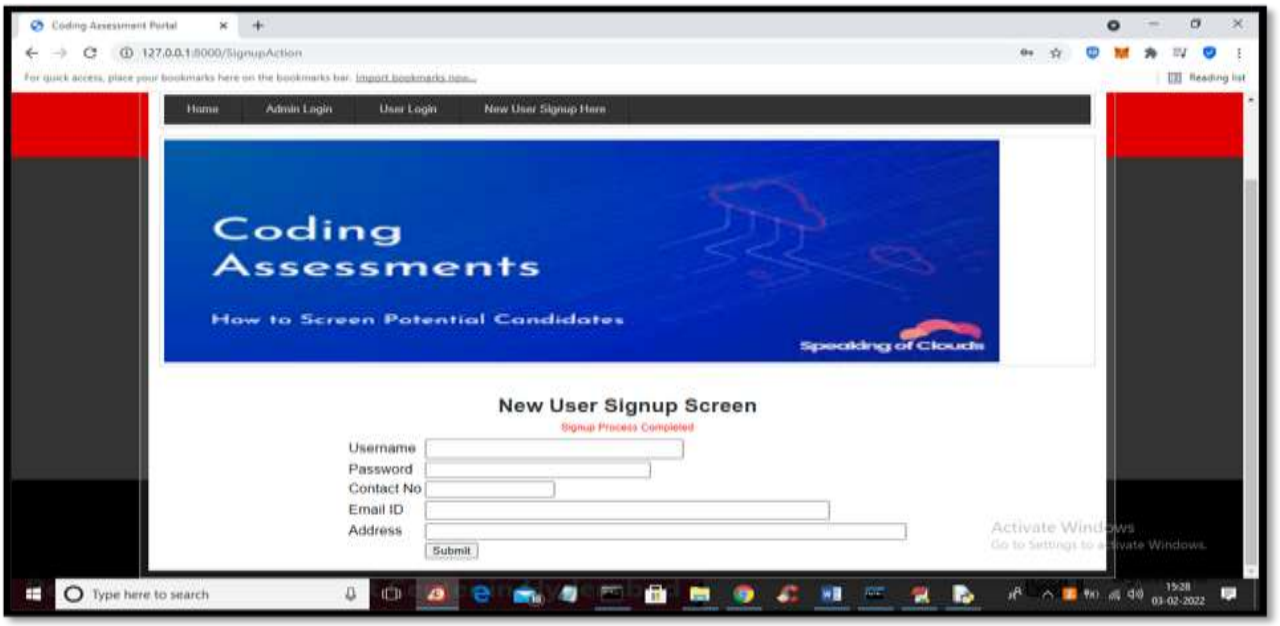

Fig 5.3 Signup Process Completed

4. In the below screen admin is login and after login will get below screen.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

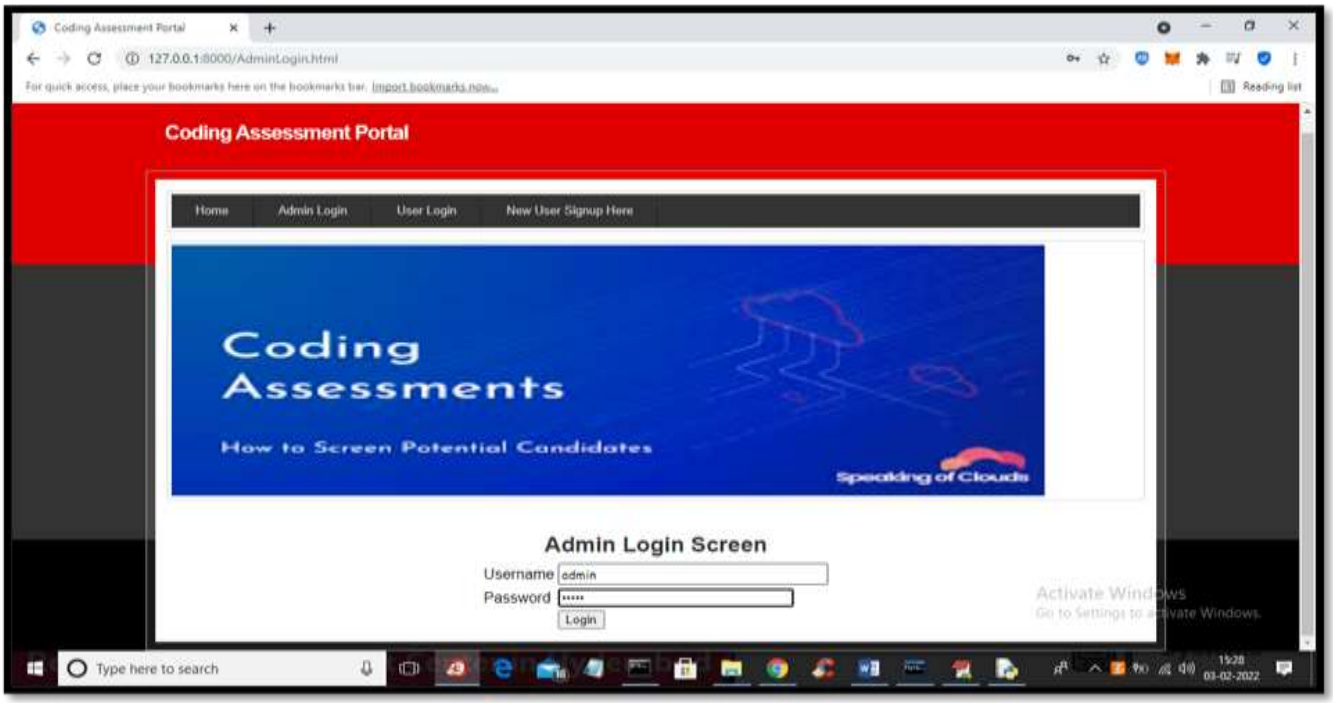

Fig 5.4:Admin Login

5. In the below screen admin can click on 'View Registered Users' link to know who are all the registered users.

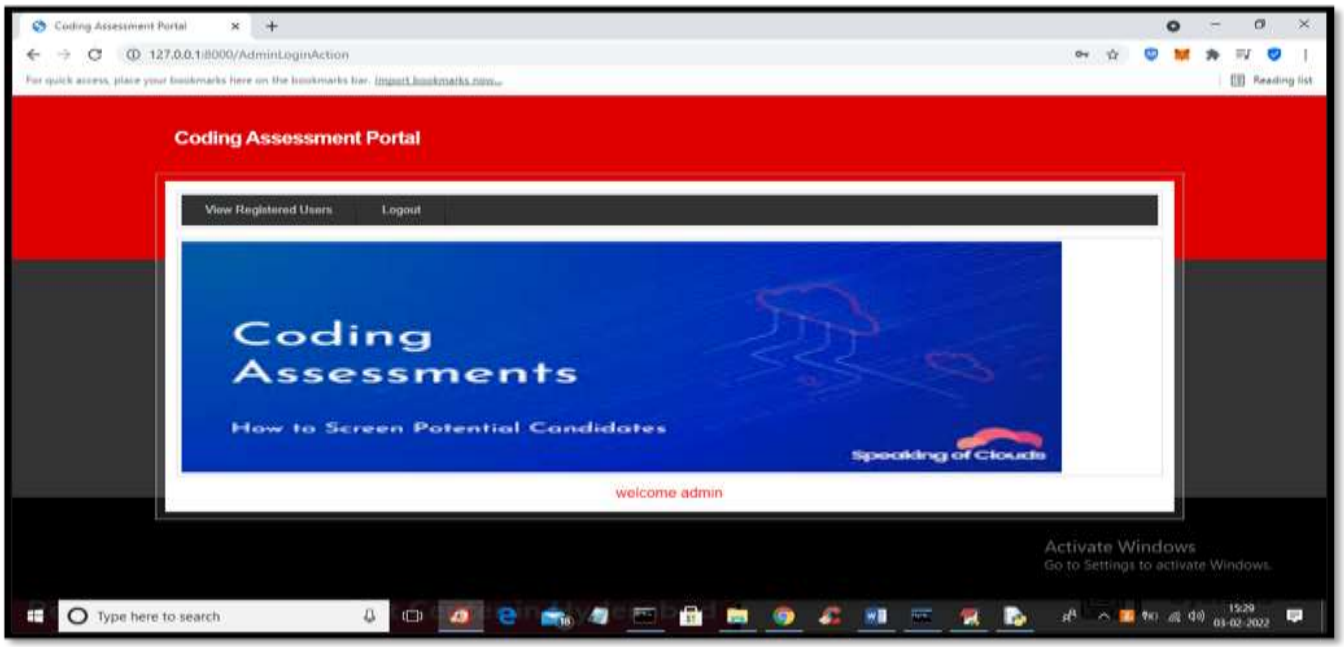

Fig 5.5 Admin portal

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

6. In the below screen admin can view all registered user details.

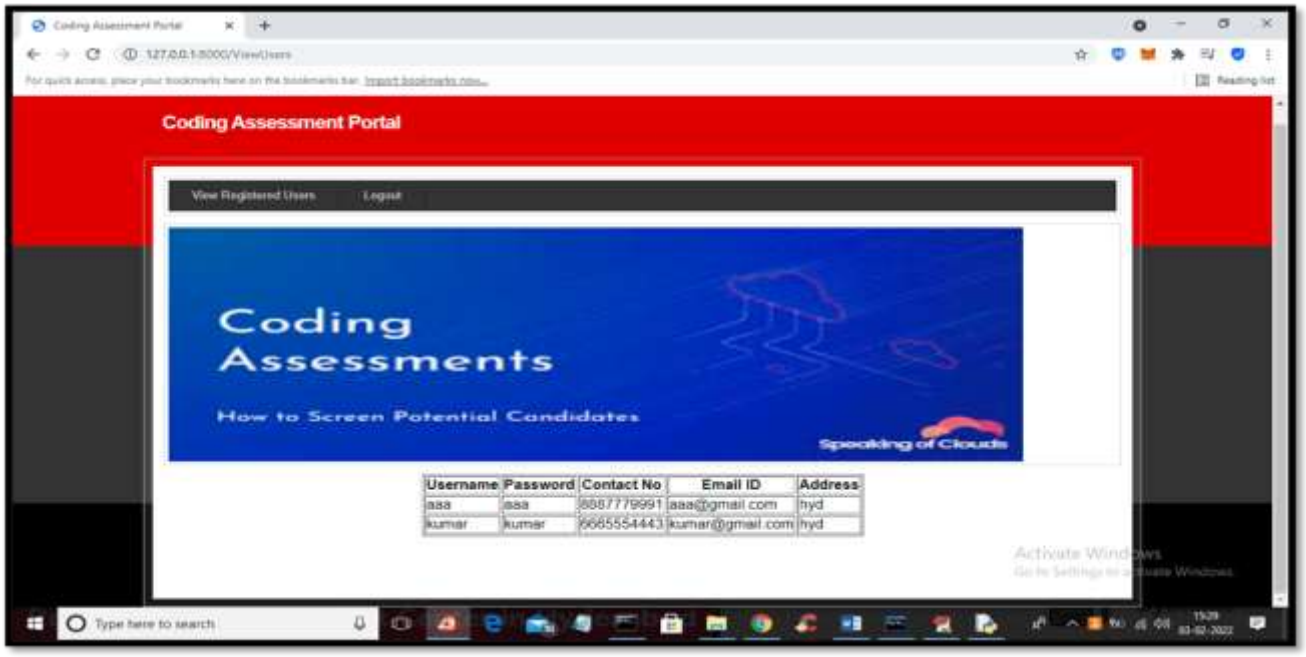

Fig 5.6 Admin viewing registered users

7. In this way the Admin can add challenges.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Fig 5.7 Add Challenges Screen

8. In this way admin can view the code which is written by the user and can give the score.

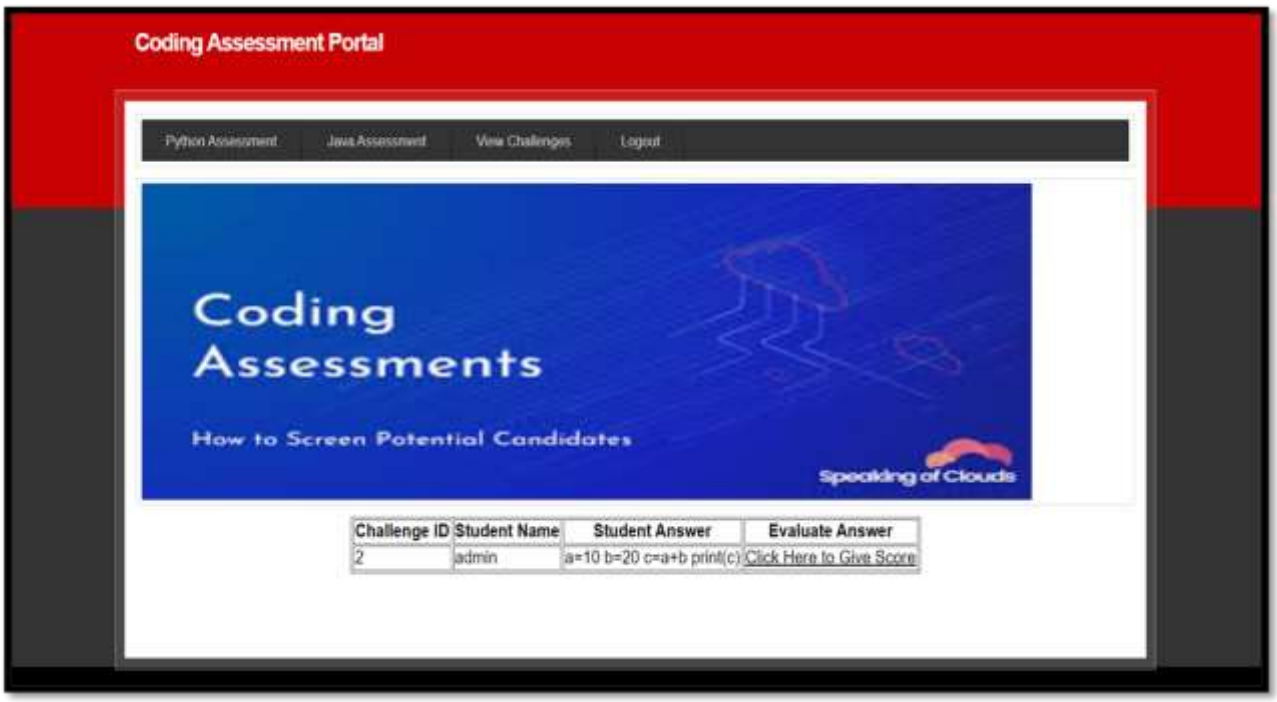

Fig 5.8 Viewing users code

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Fig 5.9 Assigning score to user

9. After giving the score a message is displayed that score is assigned successfully and then the admin can logout.

Fig 5.10 Assigning Score

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

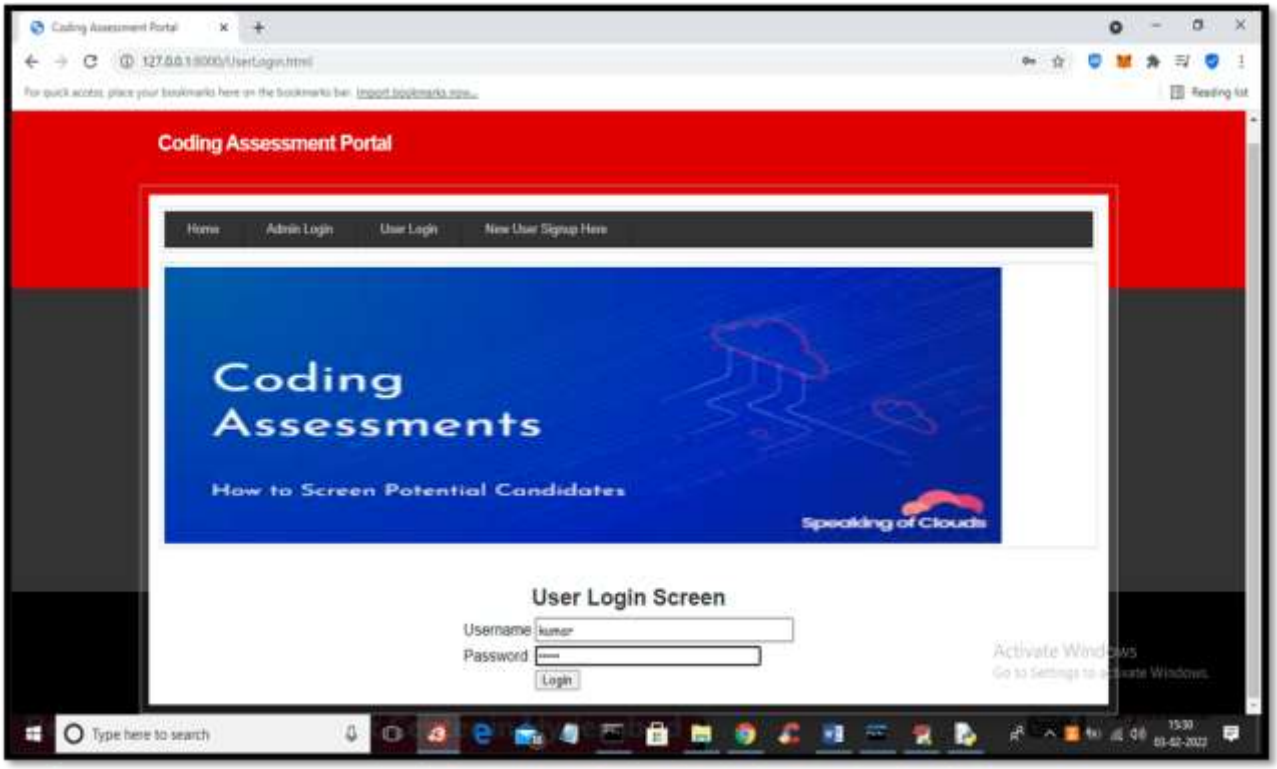

Fig 5.11 User Login

10. In the above screen user is logged in and after login will get below screen.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

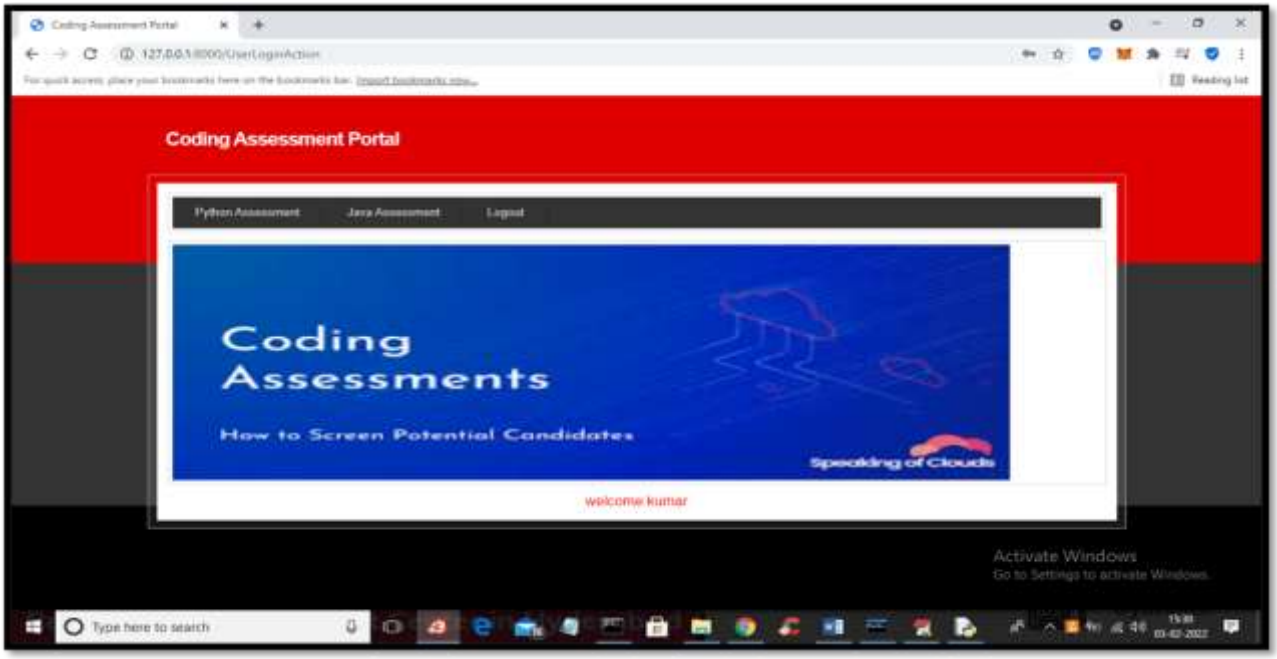

Fig 5.12 User Login tab

11. In the above screen user can click on 'Python Assessment' link to get the below screen.

Fig 5.13 Python Assessment

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

12. In the above screen in the first text area you can enter python code and in the second text area you can see output.

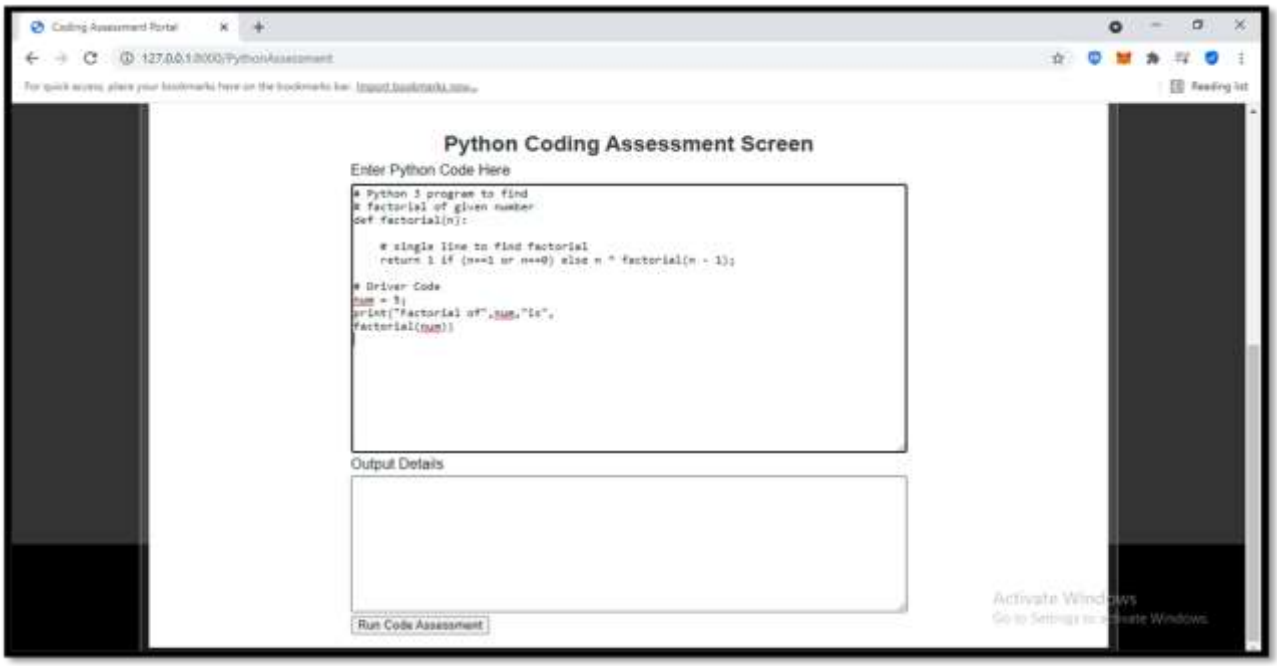

Fig 5.14 Python Assessment Screen

13. In the above screen we entered some factorial program and now press the 'Run Code Assessment' button to get the output below.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

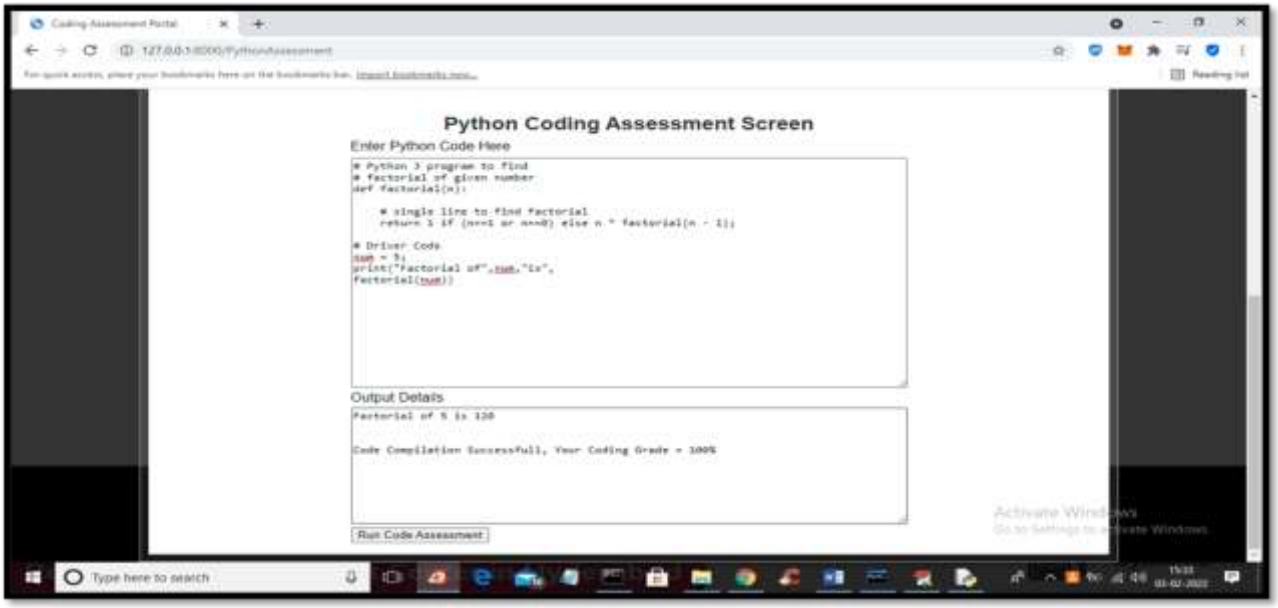

Fig 5.15 Sample program in Python

14. In the above screen in the second text area we got output as 'Factorial of 5 is 120' and we got message as Code Compilation Successful and got grade as 100% and now put some error in code and then execute again.

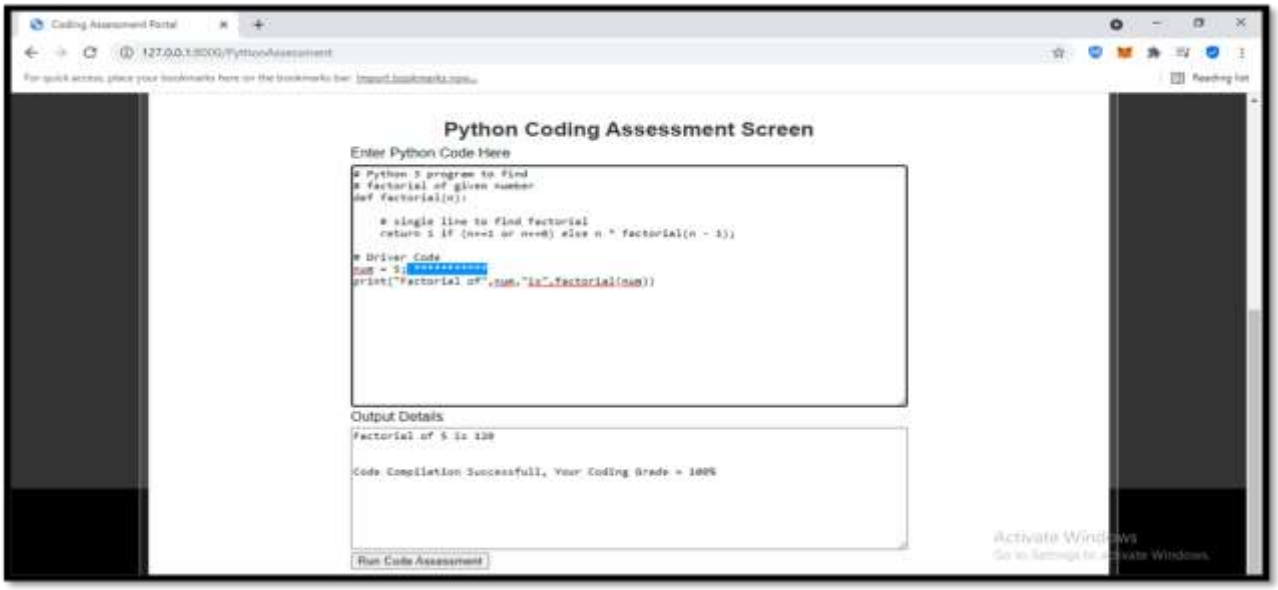

Fig 5.16 Adding garbage symbols in code

15. In above screen in blue colour text you can see we added some garbage * symbols which are not valid and press the button to get below compilation error.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

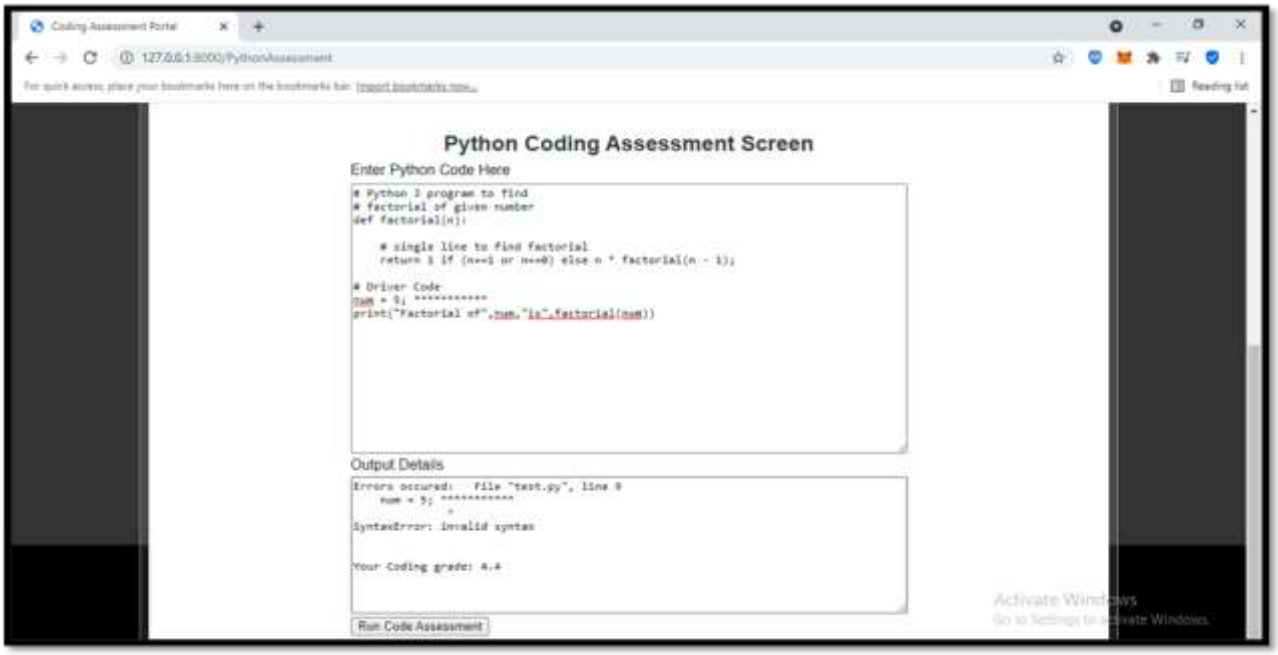

Fig 5.17 Compilation error

16. In the above screen in the second text area we got error details and we got grade as 4.4% and similarly you can test other programs also. Now click on the 'Java Assessment' link to get below screen.

Fig 5.18 Java Assessment

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

17. In the above screen in the first text area you can enter a java program and in the second text area you will see error or output details.

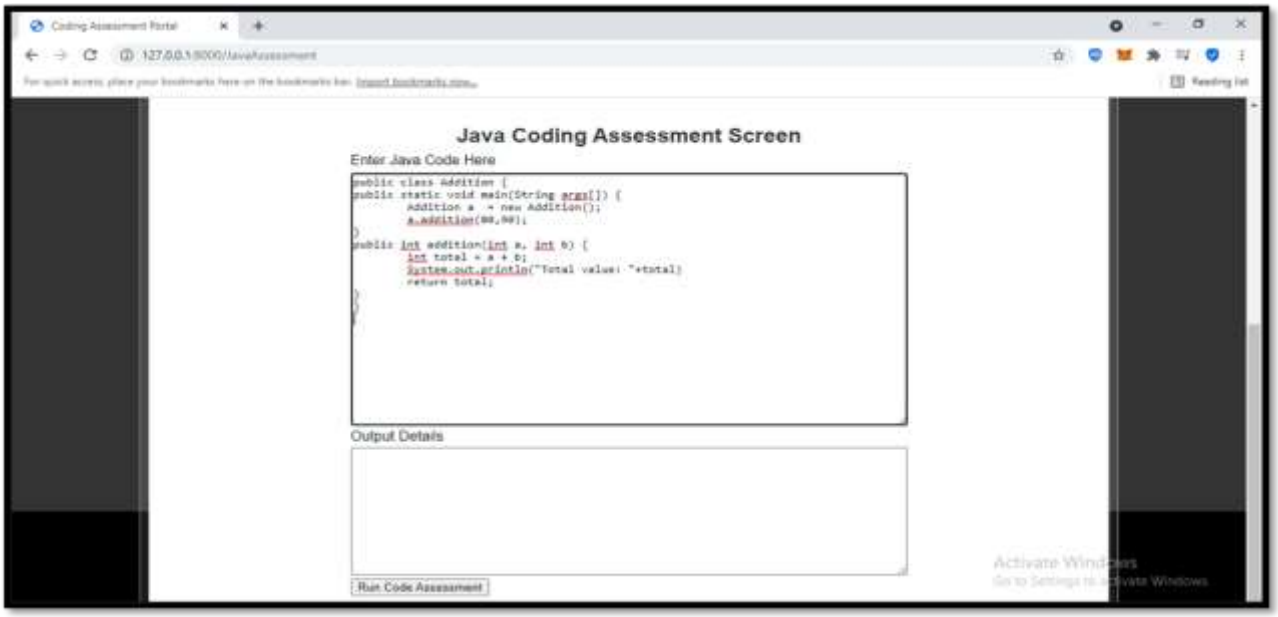

Fig 5.19 User entering java code

18. In the above screen I entered the addition of a 2 numbers program and pressed the Run Code Assessment button to get the output below.

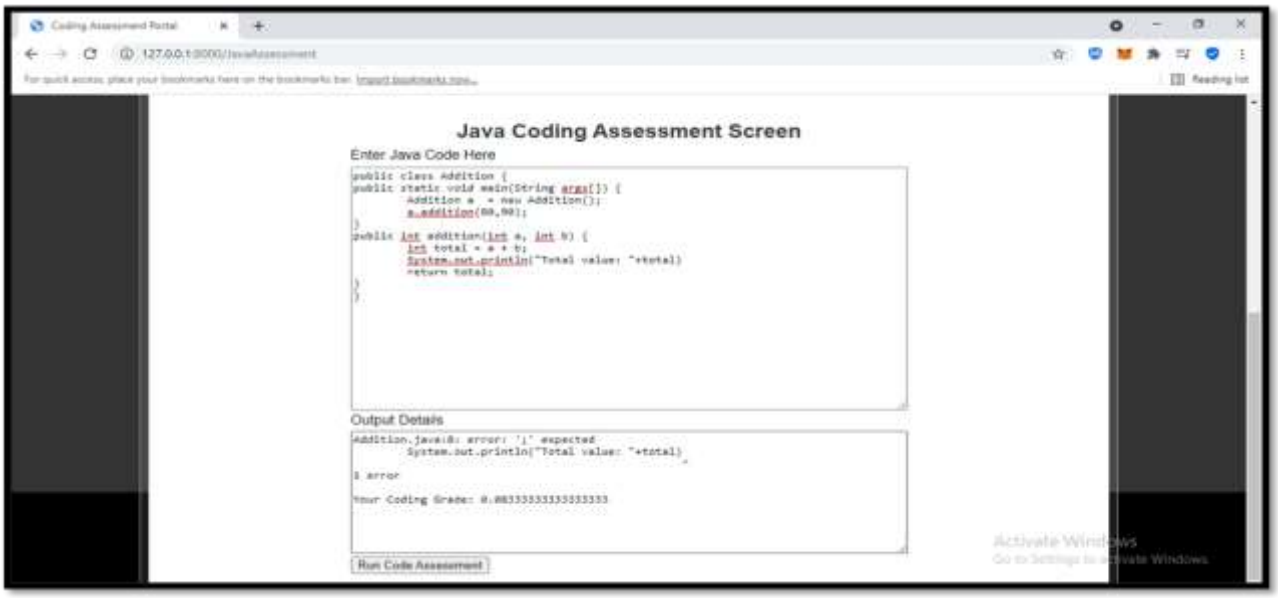

Fig 5.20 Java program

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

19. In the above screen I got an error as I forgot to put ';' after System.out. statement and now I will run the same program after putting a semicolon. The program got compiled and got additional output as 170 and got 100% as the program compiled successfully.

Fig 5.21 Program executed successfully

6.Conclusion

- The Online Coding Assessment Portal is developed using Python and Mysql fully meets the objectives of the system for which it has been developed.
- The system has reached a steady state where all bugs have been eliminated.
- The system is operated at a high level of efficiency and all the teachers and users associated with the system understand its advantage.
- The system solves the problem.
- It was intended to solve the requirement specification.

7.Future Scope

Scope of this project is very broad in terms of other manually taking exams. Few of them are:-

- This can be used in educational institutions as well as in the corporate world.
- Can be used anywhere any time as it is a web based application(user location doesn't matter).
- No restriction that the examiner has to be present when the candidate takes the test.

8.References

1. https://ieeexplore.ieee.org/abstract/document/7368995

2. https://www.youtube.com/playlist?list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF

3. https://www.youtube.com/watch?v=SDN2GixPwWw&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=2

4. https://www.youtube.com/watch?v=4H9gYuTd9b8&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=3

5. https://www.youtube.com/watch?v=Gy9OvbkCfpY&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=4

6. https://www.youtube.com/watch?v=Wb5a1pr5rnc&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=5

7. https://www.youtube.com/watch?v=jOAGrpSisf0&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=6

8. https://www.youtube.com/watch?v=bPt5bhoVFTg&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=7

9. https://www.youtube.com/watch?v=QBkNr6zPKh0&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=8

10. https://www.youtube.com/watch?v=Oau5a4wpyBg&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=9

11. https://www.youtube.com/watch?v=lraB-wB3avg&list=PL7aJ-F_dFdrF39kt5jmm6UqnFexOtYQoF&index=10

12. http://www.mediafire.com/file/kipz3ay6alpww0r/python-3.7.0-amd64.exe/file

13. http://www.mediafire.com/file/t0l3btf19qn1uce/Mysql_for_python.msi/file

14. https://www.youtube.com/watch?v=v95sxoEUQpI