A MINI PROJECT REPORT

ON

"ONLINE EXAM PORTAL"

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY

in completion of
SKILL DEVELOPMENT LABORATORY
(T.E Computer Engineering)

BY

Name of the student: Manav Chouhan Exam No :
Name of the student: Prashant Kumar Exam No :
Name of the student: Rahul Jain Exam No :

Department of Computer Engineering
Sinhgad College of Engineering, Pune-41
Accredited by NAAC with grade 'A'

YEAR 2019 - 20

CERTIFICATE

Sinhgad Technical Education Society,
Department of Computer Engineering
Sinhgad College of Engineering, Pune-41

"ONLINE EXAM PORTAL"

Submitted to

SAVITRIBAI PHULE PUNE UNIVERSITY

in completion of

SKILL DEVELOPMENT LABORATORY
(T.E Computer Engineering)

BY

Name of the student: Manav Chouhan	Exam No :
Name of the student: Prashant Kumar	Exam No :
Name of the student: Rahul Jain	Exam No :

Prof.	Prof. Sakeeb Seikh
Internal Guide	Head of Dept.
Department of Computer Engineering	Department of Computer Engineering

Dr. S.D. Lokhande
Principal
SCOE, Pune

CONTENTS

4. IMPLEMENTION & CODING

1. Methodology
2. Database schema

3 Algorithms

5. Screenshots

6. List of Database Tables

7. RESULTS & DISCUSSION

4.1 Visualization of results (Graphs, Charts, etc.)

8. CONCLUSION

REFERENCES

- **World Wide Web**

1. W3school, https://www.w3schools.com/bootstrap4
2. Geeks for geeks, https://www.geeksforgeeks.org
3. Google, https://www.google.com
4. Make a web app using python and Flask , https://aryaboudaie.com/python/technical/educational/web/flask/2018/10/17/flask.html
5. Passing plots to web page ,https://galaxydatatech.com/2018/04/01/passing-plots-web-page/
6. Quick start Flask documentation , https://flask.palletsprojects.com/en/1.1.x/quickstart/#accessing-request-data

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Prof. Sakeeb Seikh for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to staff members of SCOE for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Lastly, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this project would not be possible.

ABSTRACT

Online Examination System is web based application for technical evaluation. ExaminGo not only replace paperwork but also releases the workload of faculty. Most of e-examination system only have fixed no of question without randomization, so they have pool scalability. The proposed system has a comprehensive test engine with randomization of questions and it also allow users to give feedback of system The ExaminGo overcomes the shortcoming of existing online examination systems and has better extensibility and flexibility.

INTRODUCTION

1. Introduction

Online Examination Series (ExaminGo) is a Multiple Choice Questions (MCQ) based examination system. It provides an easy to use environment for both Test Conductors and Students appearing for Examination. The main objective of OLES is to provide all the features that an Examination System must have, with the "interfaces that don't Scare it's Users!".

2. Scope

- The main purpose of the system is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves a lot of time but also gives fast results.
- It is a cost-effective and popular means of mass- evaluation system.
- The administrator of the system prepares the tests and questions for each exam.
- The candidates can login through the client computers with their register number given to them and can take the exam.
- The questions are shuffled in a random order so that possibilities for getting questions in the same order for the students who are beside, is very less.

3. Taxonomy of ExaminGo

Users of ExaminGoare classified into three categories:

- Administrators
- Test Conductors and
- Students

1. Administrators

Administrators are responsible for management of system users, tests, results and system backup etc.

2. Faculty

Test conductors are responsible for preparing schedule of tests and questions.

3. Students

Students are the candidates who are appearing for the Exam.

4. Existing system

The Existing system of conducting examination process is manual.

Existing system is a large man power process and is difficult to implement it at different platform. It has so many problems. So we introduce a ExaminGo system, which is fully computerized. Existing system is a large man power process and is difficult to implement.

5. Disadvantages of existing system

- The existing systems are very time consuming.
- It is difficult to analyze the exam manually.
- Results are not precise as calculation and evaluations are done manually.
- Result processing after summation of exam takes more time as it is done manually.

6. Objective of proposed system

- Economic feasibility
- Time Flexibility
- Technical feasibility
- User-friendly interface

7. Limitation of Project

There are already many Online Examination System and they are directly launching their own web sites which can be accessed by the users. The limitation of this project is that it runs on a single server within the education institute or corporate world but not over internet. An Intranet application can provide much better security.

8. Organisation of Report

In this project documentation we have initially put the definition and objective of the project as well as the design of the project which is followed by the implementation and testing phases. The project has been concluded successfully and the future enhancements of the project also given in this documentation.

PROJECT PLANNING & MANAGEMENT

2.1. HARDWARE AND SOFTWARE REQUIREMENTS

- Python Version 2.7.x or above .
- Operating system : Windows 7 and Later, Linux , MacOS .
- A good integrated development environment(IDE) .
- Following python libraries –
    1. Pickle
    2. Flask
    3. Numpy
    4. Pandas
    5. Matplotlib
    6. Mysqldb

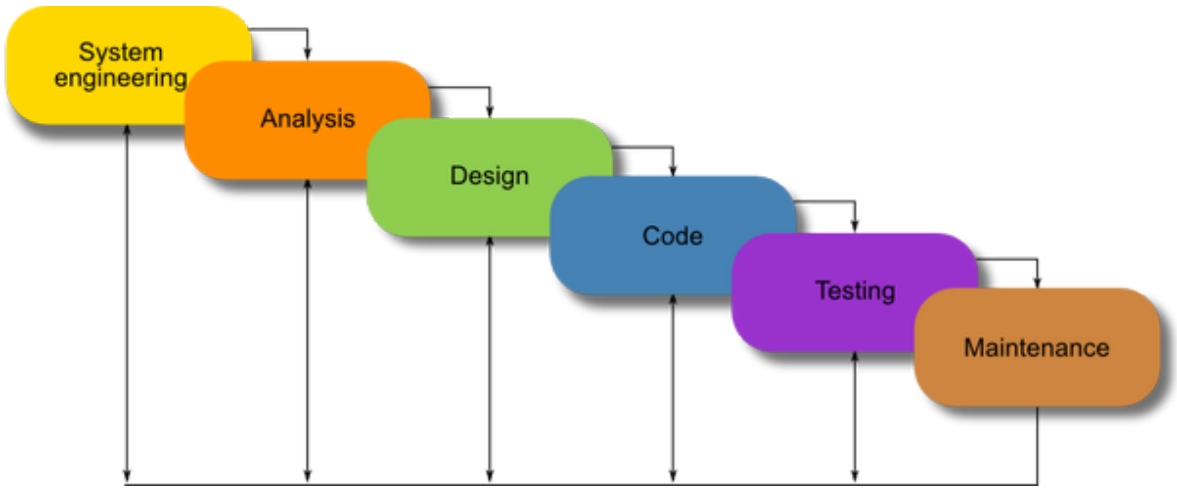
2.2. PROCESS MODEL

We have used waterfall model to complete the following project.

First, Requirement analysis and system design was done that includes discussion about objective of the project and hardware & software requirements. This also including planning about the flow of control in the project.

Implementation was carried out which includes development of smaller modules that is required to solve a part of software and then these modules were linked together to get the final project.

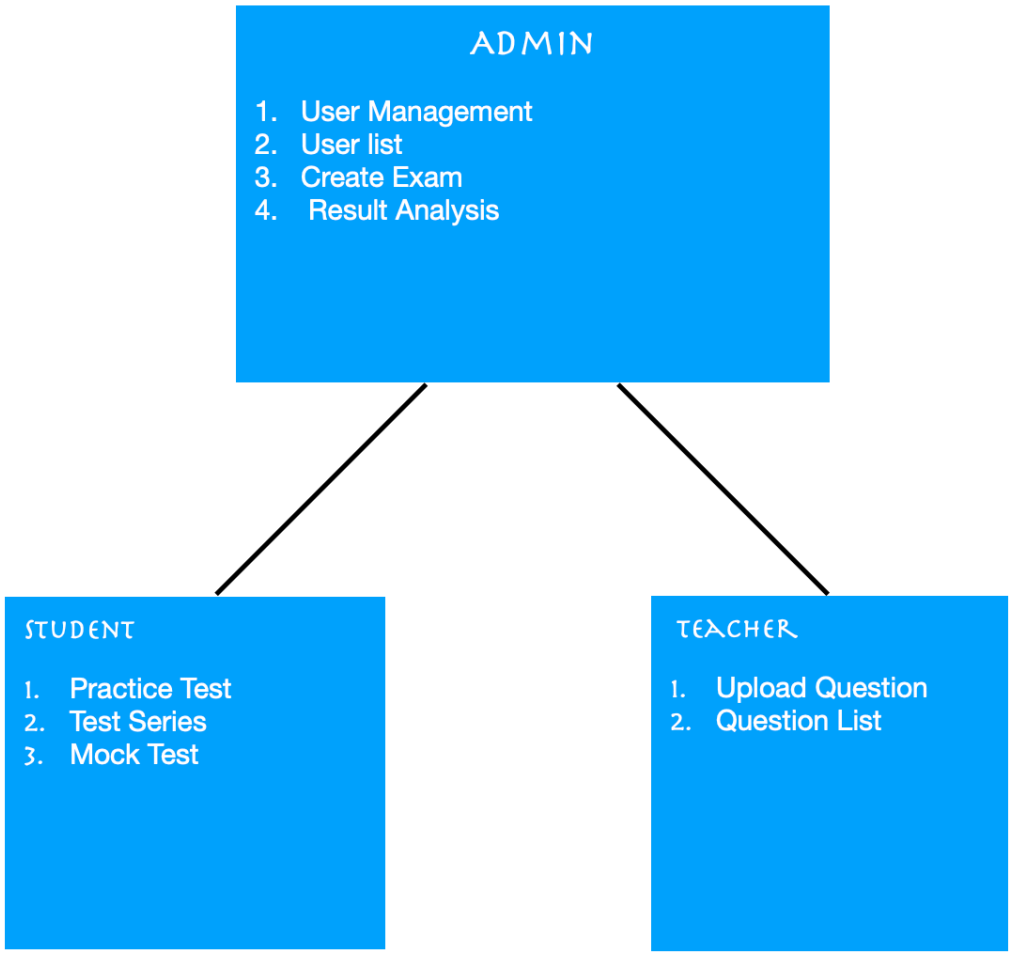
Testing was carried to check if all the objectives has been covered and dealt with properly. This also includes testing of functional and non-functional modules of the project. In case of error, the code of the particular module was rechecked and required changes was made so that it can work properly.



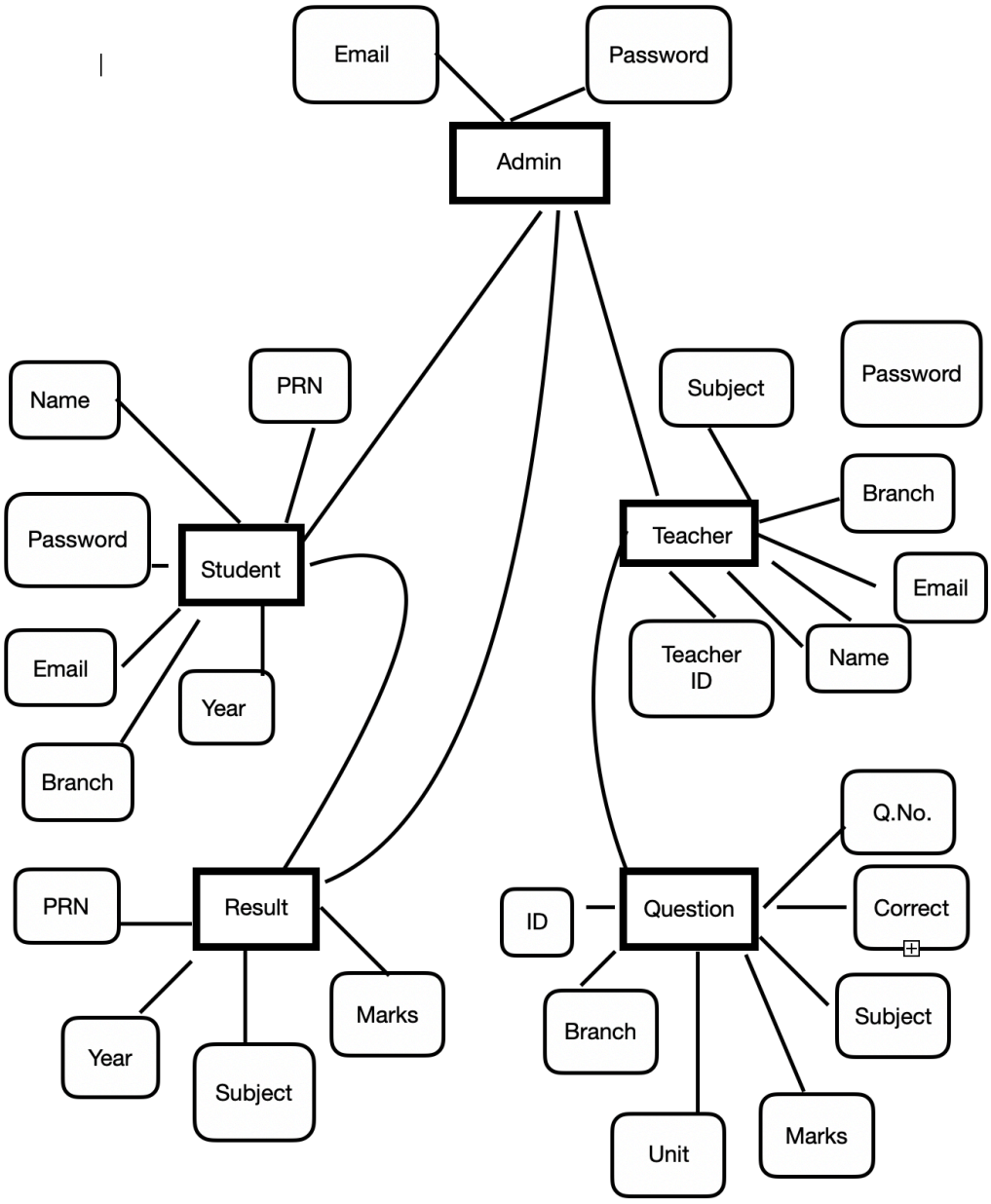
After testing phase, maintenance at regular interval was carried out at regular intervals to make the project more and more efficient and effective. This phase also includes making changes to the code if there is any changes in the requirements by the end user (if any).

ANALYSIS AND DESIGN

3.1. CLASS DIAGRAM



3.2. E – R DIAGRAM



IMPLEMENTION & CODING

1. Code

1.1. Login form

```
@app.route('/',methods=["GET","POST"])
def slogin():
    if request.method == 'POST':
    email = request.form['email']
    password = request.form['password']
    curl = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    curl.execute("SELECT * FROM users WHERE email=%s",(email,))
    user = curl.fetchone()
    curl.close()

    if user:
        if (password == user["password"]):
        session['name'] = user['name']
        session['branch'] = user['branch']
        session['email'] = user['email']
        return redirect(url_for("sdashboard"))
        else:
            error =  "Password and email not match"
            return render_template("home.html" , value = error)

    else:
        error = "User not found"
        return render_template("home.html" , value = error)
    else:
```

```
                return render_template("home.html")
```

1.2 Registration

```
@app.route('/sregister', methods=["GET", "POST"])
def sregister():
        if request.method == 'GET':
                return render_template("sregister.html")
        else:
                name = request.form['name']
                email = request.form['email']
                password = request.form['password']
                prn = request.form['prn']
                year = request.form['year']
                branch = request.form['branch']
                cur = mysql.connection.cursor()
                cur.execute("SELECT * FROM teacher WHERE email=%s",
[email])
                user = cur.fetchone()
    if user:
      error = ("User exist")
      return render_template("sregister.html" , value = error)
   else:
      cur = mysql.connection.cursor()
      cur.execute("INSERT INTO users (name, email, password , prn , year ,
branch) VALUES (%s,%s,%s,%s,%s,%s)",(name,email,password,prn,year,branch))
      mysql.connection.commit()
      session['name'] = request.form['name']
      session['email'] = request.form['email']
      flash("User Created...")
```

```
            return redirect(url_for('sregister'))
```

1.3 Question Upload

```
        @app.route('/tupload', methods=["GET", "POST"])
def tupload():
    if request.method == 'POST':
        file = request.files['file']
        book = pd.read_csv(file)
        subject = request.form['subject']
        unit = request.form['unit']
        branch = request.form['branch1']
        for row in range(len(book)):
            rows = (book.iloc[row][0],book.iloc[row][1],book.iloc[row]
[2],book.iloc[row][3],book.iloc[row][4],book.iloc[row][5],book.iloc[row]
[6],subject,unit,branch)
            cur = mysql.connection.cursor()
            cur.execute("INSERT INTO
question(question,option1,option2,option3,option4,correctoption,marks,subject,unit,
branch) VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",rows)
            mysql.connection.commit()
        flash( "File Uploaded...")
        return redirect(url_for('tupload'))
    else:
        return render_template("tupload.html")
```

1.4 Exam

```
@app.route('/mock',methods= ["POST","GET"])
def mock():
    if request.method == 'GET':
        branch = session['branch']
        curl = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        curl.execute("select distinct subject from question where branch = %s",
[branch])
        data = curl.fetchall()
        curl.close()
        data1 = len(data)
        return render_template("mock.html" , value = data , len = data1)



@app.route('/mock2',methods= ["POST","GET"])
def mock2():
    if request.method=="POST":
        leng=5
        sub = request.form['sub']
        curl = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        curl.execute("select * from question where subject=%s order by rand() limit
%s",(sub,leng))
        data = curl.fetchall()


        i=1
```

```
        for item in data:
            item.update({'srno':i})
            i=i+1
            if i>len(data):
                break
        curl.close()

        return render_template("exam.html", value=data , length=leng)
```

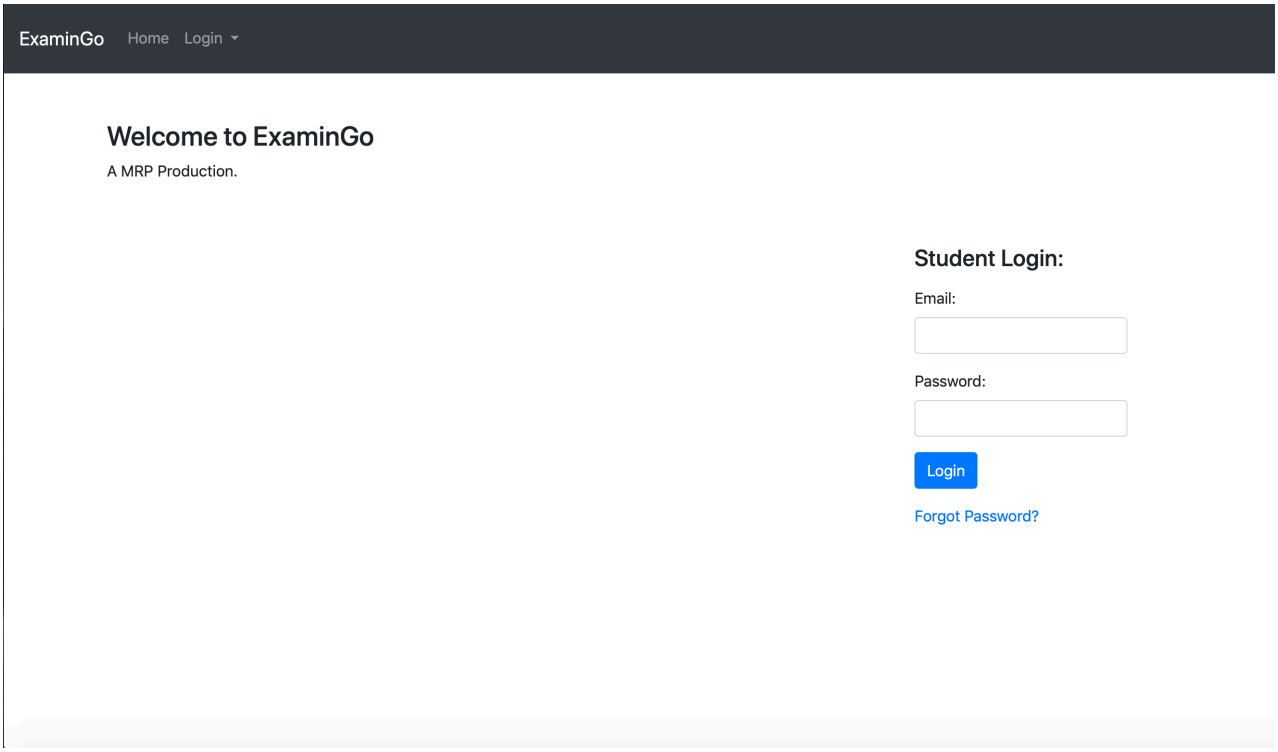
1.5 Result

```
@app.route('/result', methods=['POST'])
def result():
    data=request.json
    leng=len(data)
    option=[]
    marks=0
    for i in range(0,leng):
        option.append(data[i][0])
        if int(option[i])==int(data[i][1]):
            marks=marks+1
    result=int((marks/leng)*100)
    for i in range(0,leng):
        curl1 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        curl1.execute("insert into result(id,selected,correct) values (%s,%s,%s)",(data[i]
[2],data[i][0],data[i][1]))
        mysql.connection.commit()
    curl1.close()
    return jsonify({'marks':result , 'urll': url_for('displayresult',name=result)})
```

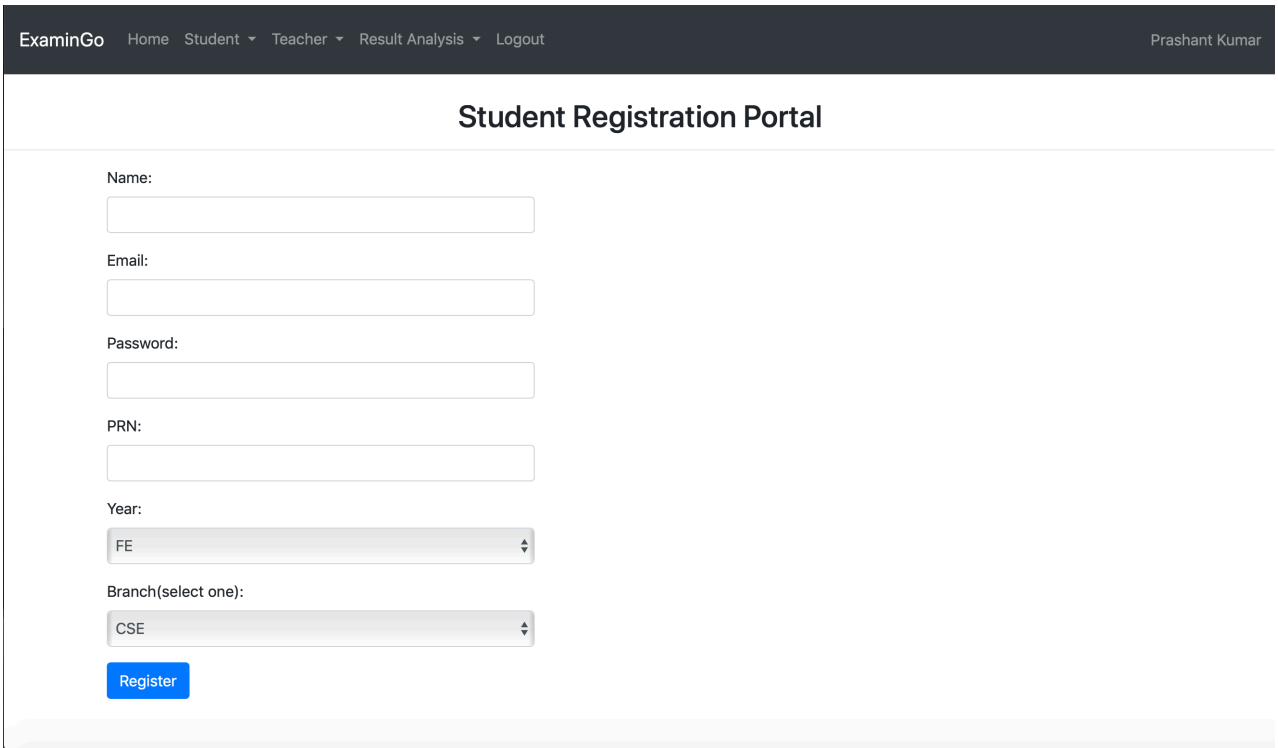
```
@app.route('/displayresult/<string:name>', methods=['GET','POST'])
def displayresult(name):
    curl = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    curl.execute("select * from result")
    data = curl.fetchall()
    curl.close()

    curl1 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    curl1.execute("select * from question where id in(select id from result)")
    data1 = curl1.fetchall()
    lengt=len(data1)
    curl1.close()

    curl2 = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    curl2.execute("delete from result")
    curl2.close()
    mysql.connection.commit()


    return
render_template("result_ques.html",value=data1,value1=data,length=lengt,result=na
me)
```

2. Screenshots:

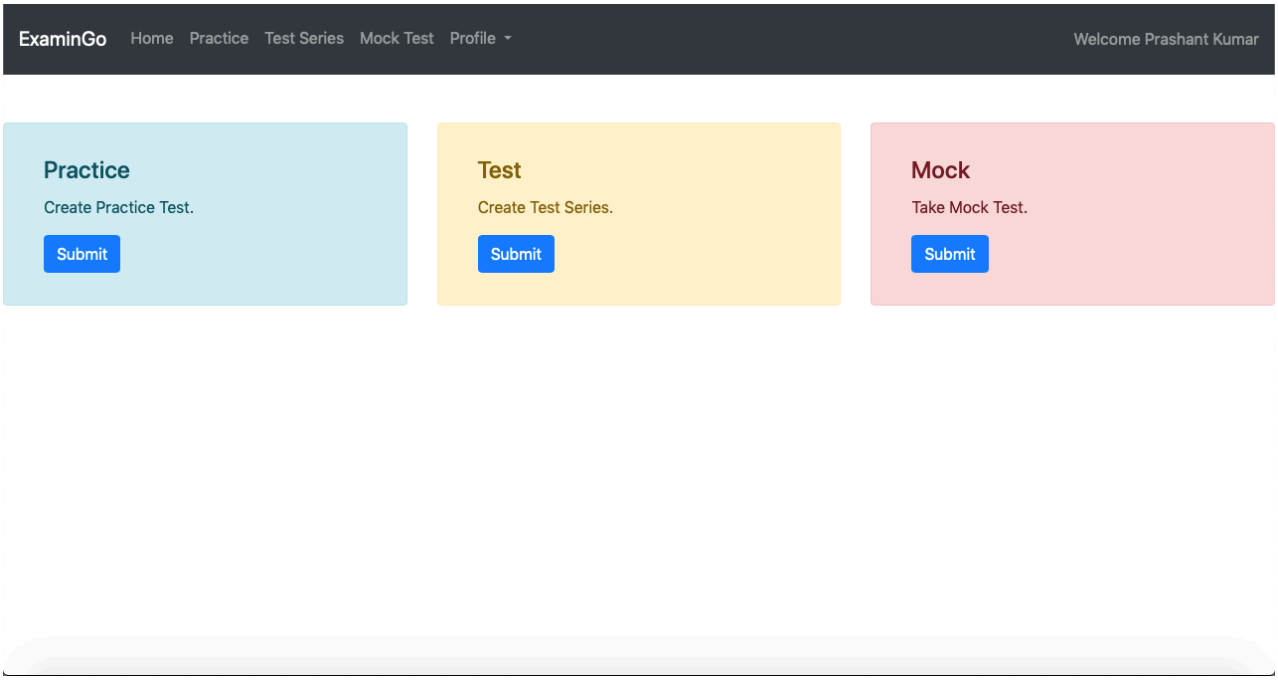
2.1 Home Page



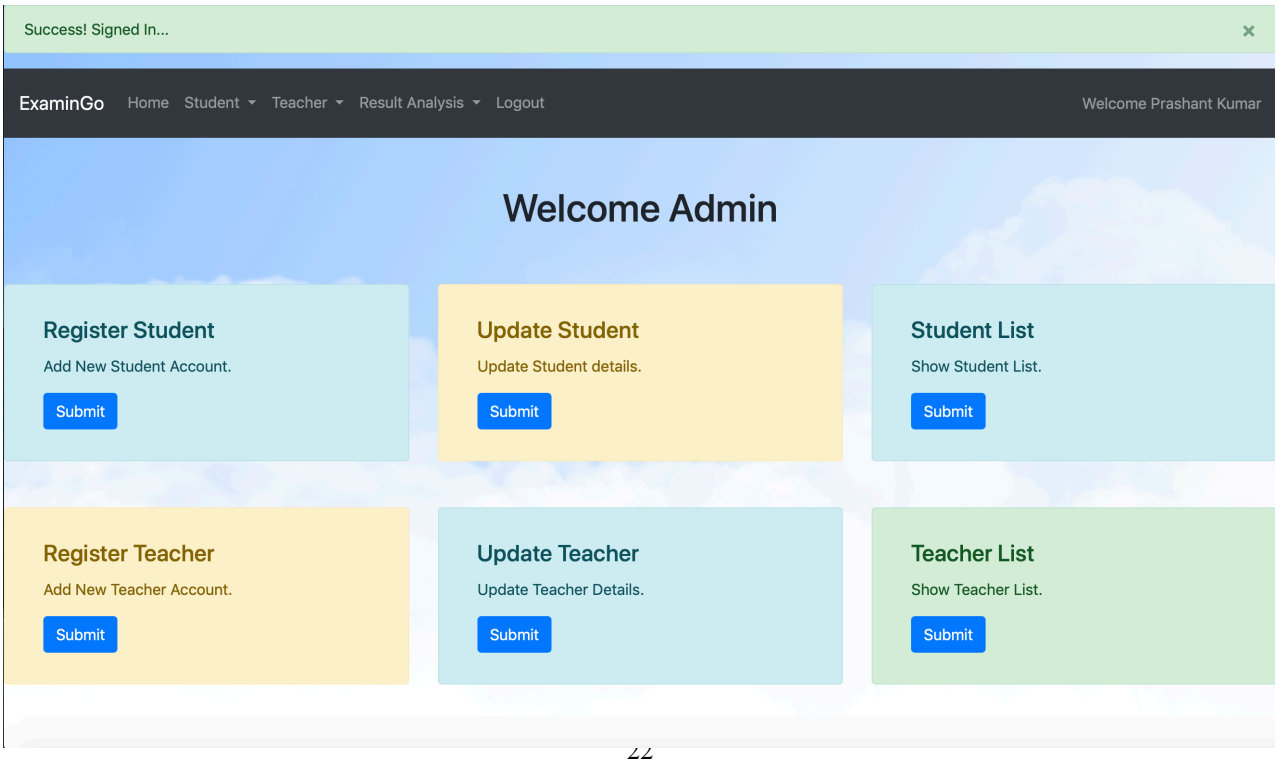
2.2 Student Registration



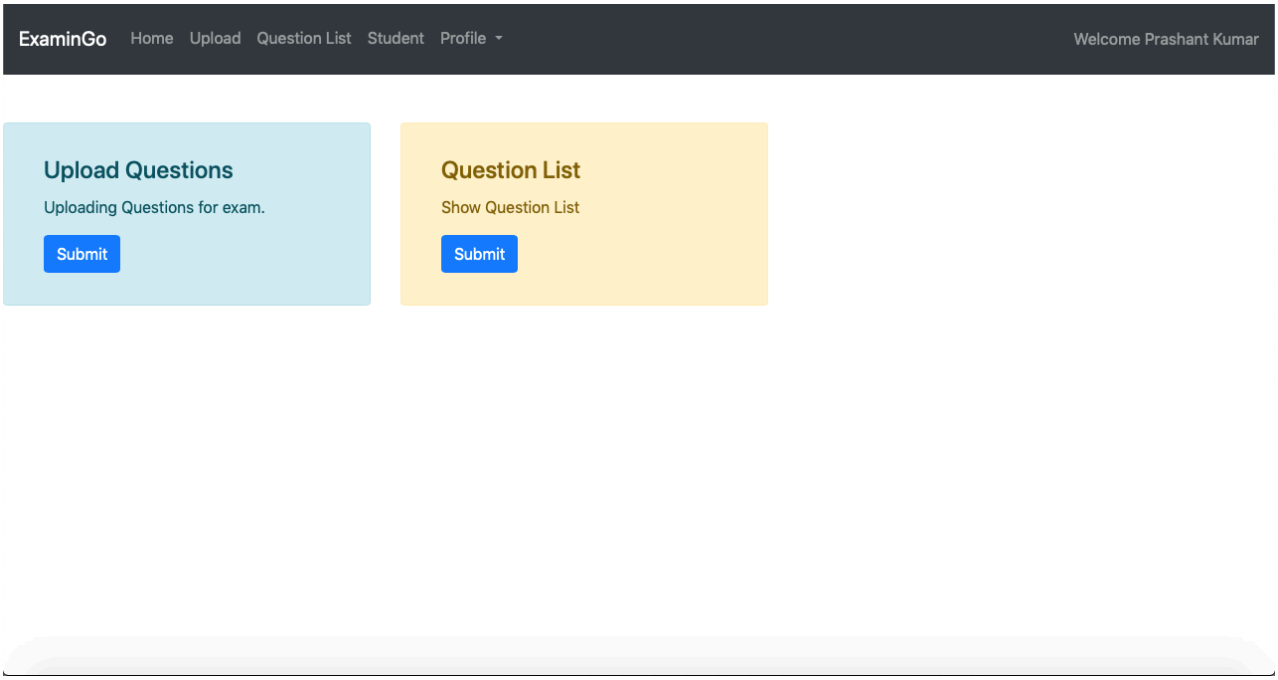
2.3 Student Dashboard



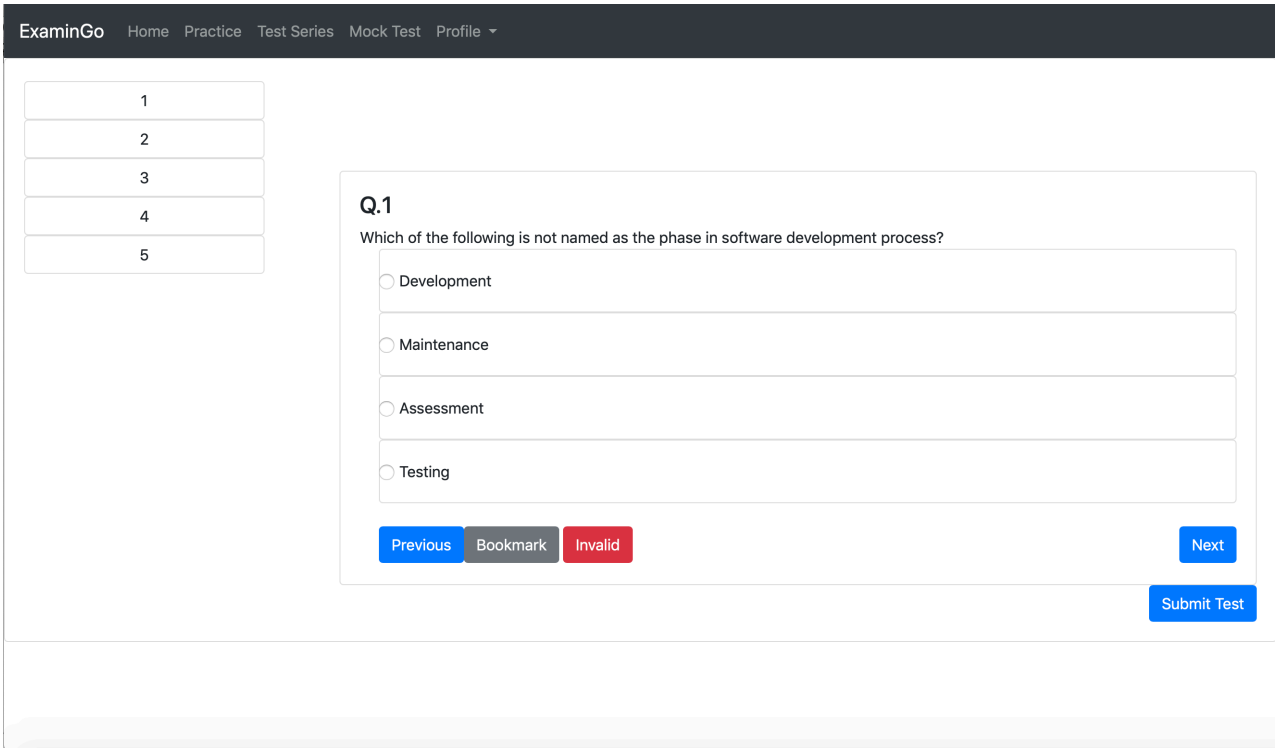
2.4 Admin Dashboard



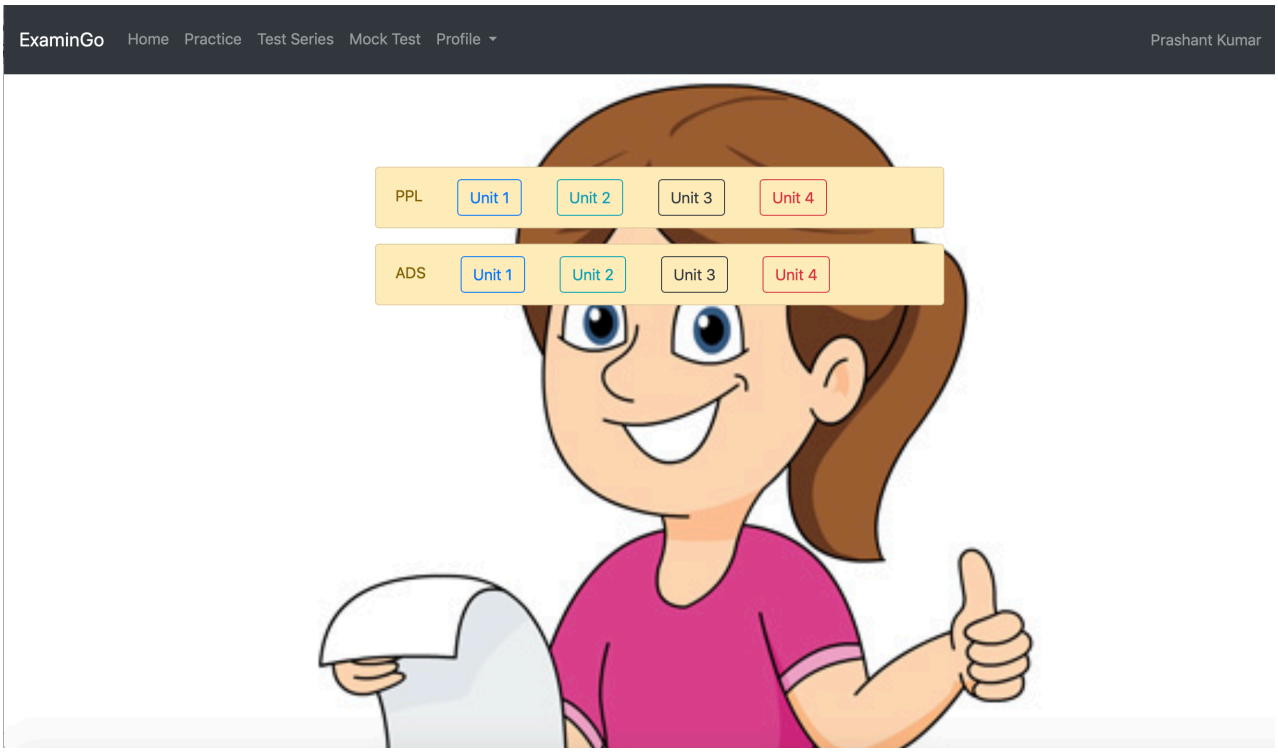
2.5 Teacher Dashboard



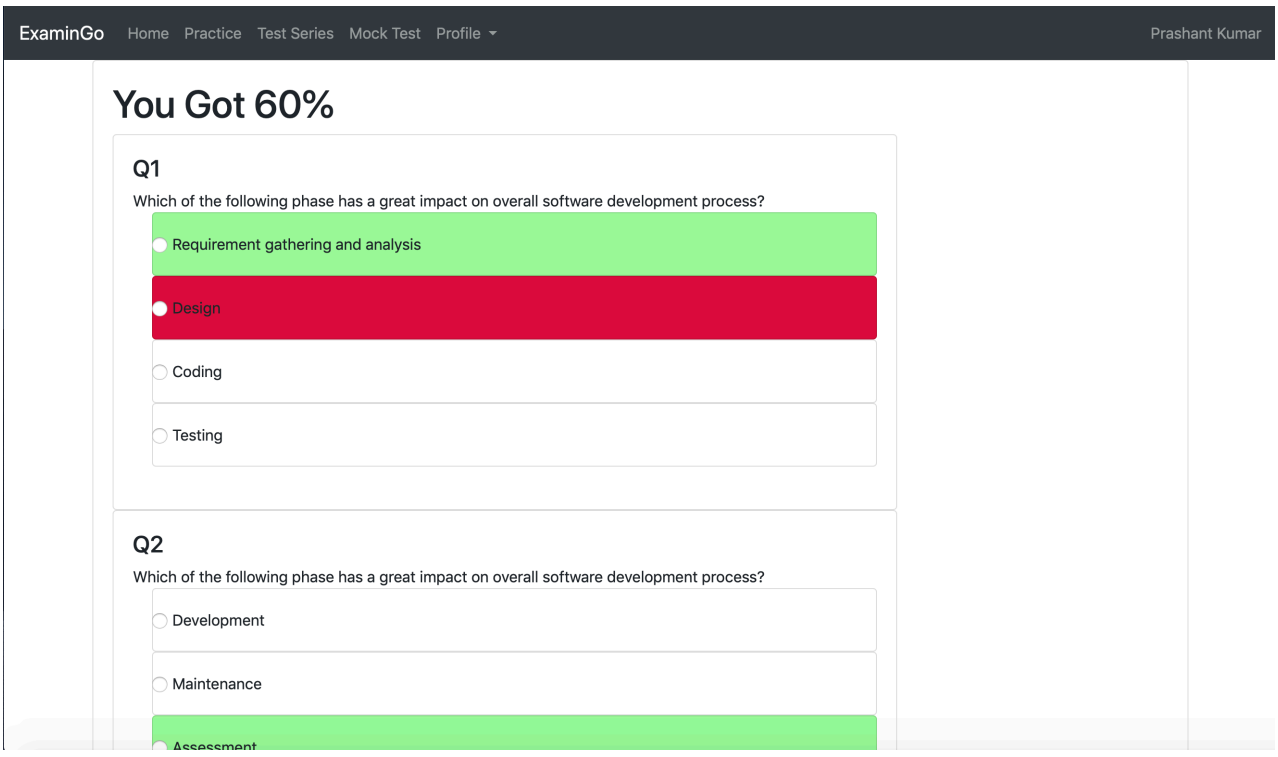
2.6. Exam



2.7 Practice Test



2.8 Result



List of Databases Tables

1. Users

Field	Type	Null	Key	Default	Extra
Id	int(10)unsigned	No	Primary key	Null	auto_increment
Name	varchar(20)	yes		Null	
Email	varchar(20)	yes		Null	
Password	varchar(20)	yes		Null	
Prn	varchar(20)	yes		Null	
Year	varchar(4)	yes		Null	
Branch	varchar(10)	yes		Null	

2. Teachers

Field	Type	null	Key	Default	Extra
Id	int(10)undigned	No	Primary key	Null	auto_increment
Name	varchar(20)	yes		Null	
Email	varchar(20)	yes		Null	
Password	varchar(20)	yes		Null	
Teacher id	varchar(20)	yes		Null	
subject	varchar(20)	yes		Null	
Branch	varchar(20)	yes		Null	

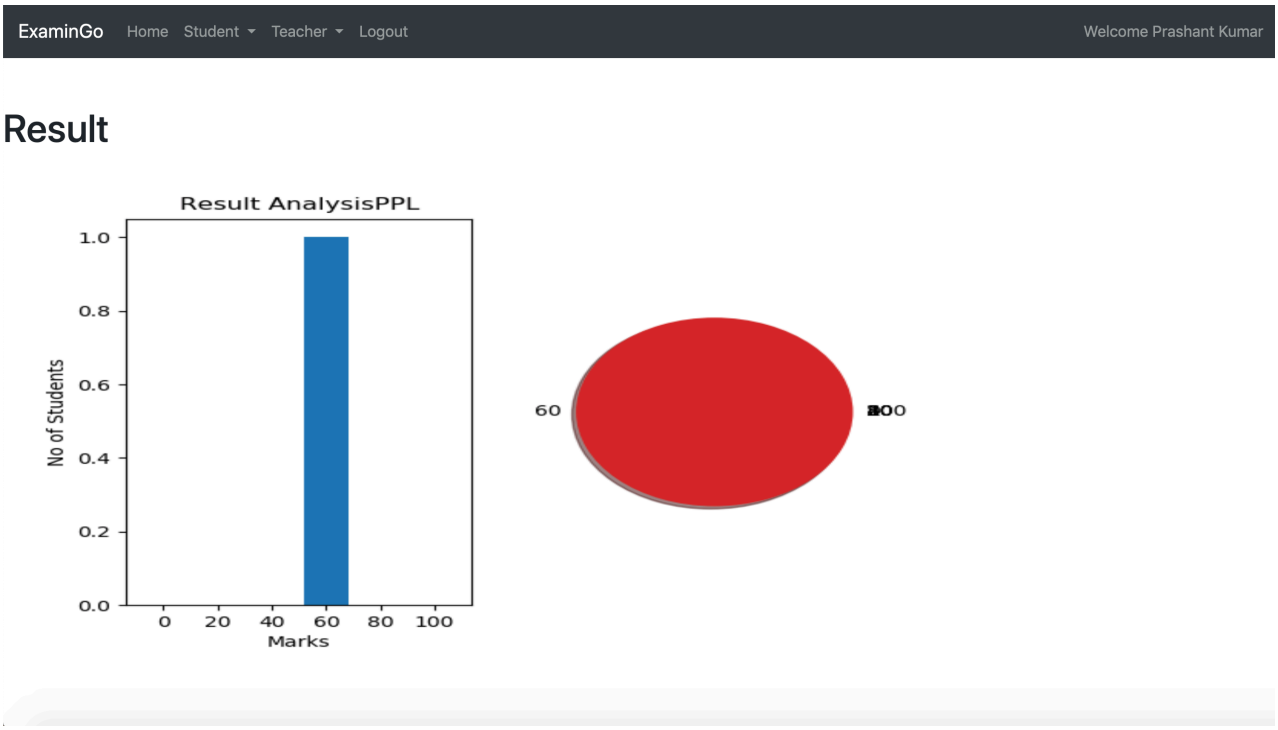
3. Question Upload

Field	Type	null	Key	default	Extra
question	int(10)unsigned	No	Primary key	null	auto_incrememt
Question	varchar(20)	yes		null	
Option 1	varchar(20)	yes		null	
option 2	varchar(20)	yes		null	
Option 3	varchar(20)	yes		null	
Option 4	varchar(20)	yes		null	
Correct option	varchar(20)	yes		null	
Marks	varchar(20)	yes		null	
Subject	varchar(20)	yes		null	
Unit	varchar(20)	yes		null	
Branch	varchar(20)	yes		Null	

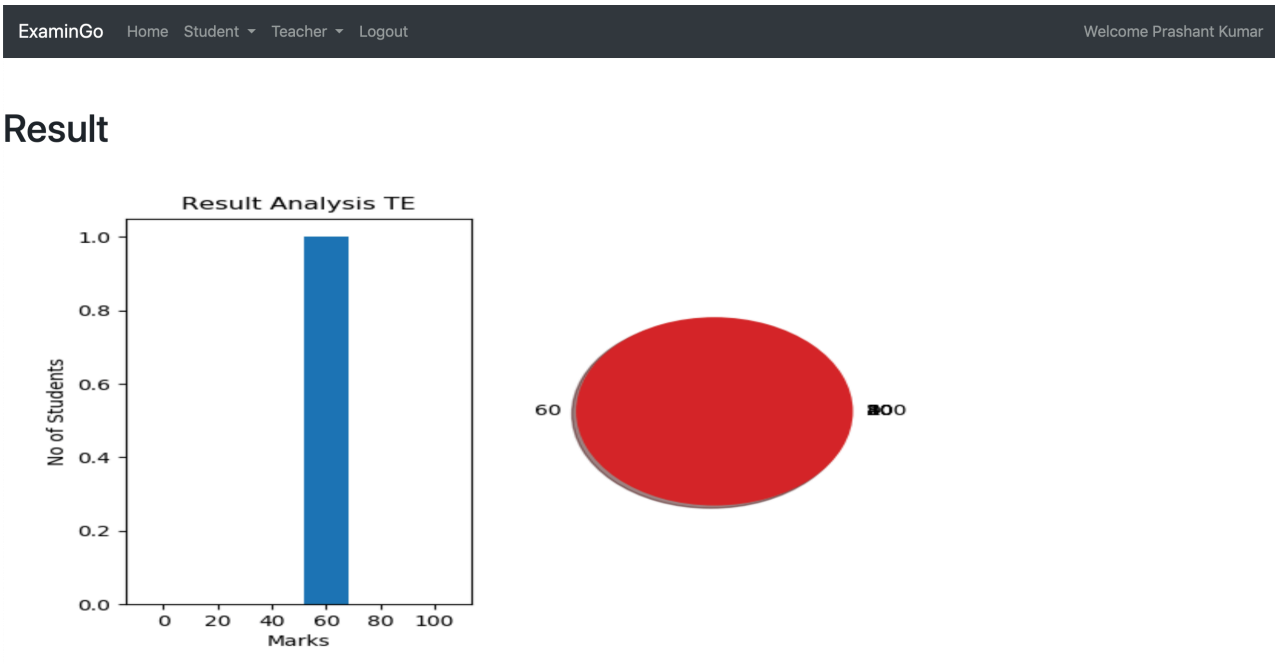
RESULTS & DISCUSSION

5.1. VISUALISATION OF RESULT

Result analysis subject wise:



Result Analysis Year Wise :



CONCLUSION

Online Examination System (ExaminGo) is a web application. The key concept is to minimize the amount of paper and convert all forms of documentation to digital form. It can observe that the information required can be obtained with ease and accuracy in the computerized system. The user with minimum knowledge about computer can be able operate the system easily. Thesystem also produces brief result
required by the management.