



RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(Catering the Educational Needs of Gifted Rural Youth of A.P)

R.K Valley, Y.S.R Kadapa (Dist) -516330

(MARKS MANAGEMENT SYSTEM)

Supervised by

MR VINOD KUMAR

Assistant Professor

Department of Computer Science Engineering

Submitted by

TEAM : TECH TRIOS

L.GAYATHRI R170839

P.VARSHA R171062

P.NIKHITHA R170283

Department of Computer Science Engineering

This Project report has been submitted in fulfillment of the requirements for the Degree of Bachelor of Technology in Minor project.

Rajiv Gandhi University Of Knowledge
Technologies
IIIT,R.K.Valley,Y.S.R. Kadapa(Dist)-516330



CERTIFICATE

This is to certify that the project work titled “STUDENT MARKS MANAGEMENT SYSTEM” is a bonafied project work submitted by L.Gayathri(R170839), P.Nikhitha(R170283),P.Varsha(R171062) in the dept of CSE in partial fulfillment of Requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering for the year 2021-2022 carried out the work under the Supervision.

GUIDE

K.VINODKUMAR

HEAD OF THE DEPARTMENT

P.HARINADHA

-

ABSTRACT

Marks management system is a web application which displays the marks obtained by the students. It is a software which is helpful for students as well as the school authorities. In the current system all the activities are done manually ,it is very time consuming and costly.Our Marks Management System deals with the various activities related to the students.

Table Of Contents

1.Introduction.....	4
1.1 Purpose.....	4
1.2 Definitions,Acronyms and Abbreviations.....	4
1.3 Project Scope.....	4
1.4 References.....	4
2.Overall Description.....	5
2.1 Product Perspective.....	5
2.2 Product Features.....	5
2.3 User classes and characteristics.....	5
2.4 Design and Implementation constraints.....	5
2.5 Assumptions and Dependency.....	5
3.System Features.....	6
3.1 Functional Requirements.....	6
4.External Interface Requirements.....	6
4.1 User Interfaces.....	6
4.2 Hardware Interfaces.....	7
4.3 Software Interfaces.....	7
5.Non-Functional Requirements.....	7
5.1 Availability.....	7
5.2 Safety Requirements.....	7
5.3 Security Requirements.....	7
5.4 Hardware Constraints.....	7
5.5 Efficiency.....	7
6.System Design.....	8
6.1 DFD Diagrams.....	8
6.2 ER Diagrams.....	9
6.3 UML Diagrams.....	9
7.Output.....	10
7.1 Login Page.....	10
7.1.1 Faculty Home Page.....	11
7.2 Marks Display.....	11
7.3 Marks Update.....	12
7.4 All Students Marks Display.....	12

1)Introduction

This document aims at defining overall software requirement for MARKS MANAGEMENT SYSTEM. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in

developing/ testing/implementing /using this product.

1.1 Purpose

This specification document describes the capabilities that will be provided by software application STUDENT MARKS MANAGEMENT SYSTEM. It also states the various constraints by which the system will abide. The intended audience for this document are the development team, testing team and end users of the product.

1.2 Definitions, Acronyms and Abbreviation

SRS - Software Requirement Specification

IEEE- The Institute of Electrical and Electronic Engineers

DFD - Data Flow Diagram

DBMS – Database Management System

OS – Operating System

HTML – Hypertext Markup Language

CSS – Cascading Style Sheet

CFD – Context Flow Diagram

ER – Entity Relationship

1.3 Project Scope

The application will manage the information about various students enrolled in different subjects, the marks will be allotted to every student in every subject by the teacher. The Administrator will manage the usernames and passwords of every user (students, faculty) Student will able to see the marks obtained in various subjects. The application will greatly simplify and speed up the result preparation and management process.

1.4 References

- * IEEE 830-998 standard for writing SRS document.
- * Software engineering by Roger S. Pressman.
- * Wikipedia and Software engineering by KK Agrawal.

2) Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

2.1 Product Perspective

The application will be windows based, self-contained and independent software product. Front end client Application (with data/entry/update/delete/view and reporting facility)

2.2 Product Features

- 1) Information about the various Users.
- 2) Information about marks offered in various subjects.
- 3) Marks obtain by Students in different subjects.
- 4) Generation of Reports.
- 5) Marks update by the teacher

2.1 User Classes and Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

2.2 Design And Implementation Constraints

Users at university will have to implement a security policy to safeguard the marks related information being modified by unauthorized users (by means of gaining access to the backend database).

2.3 Assumptions And Dependency

- 1) The number of subjects to be taken by the student in each semester does not change.
- 2) The subject types do not change.
- 3) The number of semesters does not change.
- 4) The users have sufficient knowledge of computers.
- 5) The users know the English language, as the user interface will be provided in English.

3. System Features

This section contains the software requirements to a level of detail sufficient to enable designers to design the system and testers to test that system.

3.1 Functional Requirements

Depending upon the user role student/teacher will be able to access only the specific page of the system.

- 1) Login facility for enabling only authorized access to the system.
- 2) Administrator will be able to modify /add/delete information about different students that are enrolled for different subjects.

4. External Interface Requirements

4.1 user Interfaces

The following screens will be provided:

Login screen: This will be the first screen that will be displayed. It allows user to access different screens based upon the user role.

Various fields available on this screen will be

User id: alphanumeric of length up to 20char.

Password: alphanumeric of length up to 20char.

Role will have the following

Values: Administrator, Student , Faculty

a) Student Interface:

This screen will be accessible only to the Student.

It will allow the student to view his/her details and marks in different subjects.

Various fields available on these screen are:

Student ID : alphanumeric string of length up to 20 char. Student Name: only alphabetic letters and length up to 40 chars. Student Mail ID: of the format stud~~e~~ n@g_mail om

Marks : In different subjects.

b) Faculty Interface :

This screen will be accessible only to the Teacher. It will allow the teacher to enter the Student marks according to students Id Numbers. which the user wants to access the marks information.

Various fields available on these screen are:

A navigation bar which consists of Home, Update Marks, All student marks, Logout.

4.2 Hardware Interfaces

- 1) Intel p4 processor with minimum 2GHz speed.
- 2) RAM: Minimum 1GB
- 3) Hard Disk: min 1GB

4.3 Software Interfaces

- 1) PyCharm
- 2) DB Server : XAMPP Server
- 3) OS : any operating system and all browsers.

5. NON-FUNCTIONAL REQUIREMENTS

5.1 Availability :

The website is available for 24 hours a day. It is always available for the use

5.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure ,therefore it is required to take the database backup.

5.3 Security Requirements

We are going to develop a secured database for the university. Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can able to modify data delete append etc.. All other users other than staff only have the rights to retrieve the information about database.

5.4 Hardware Constraints

The system requires a database in order to store persistent data. The data base should have back up capabilities.

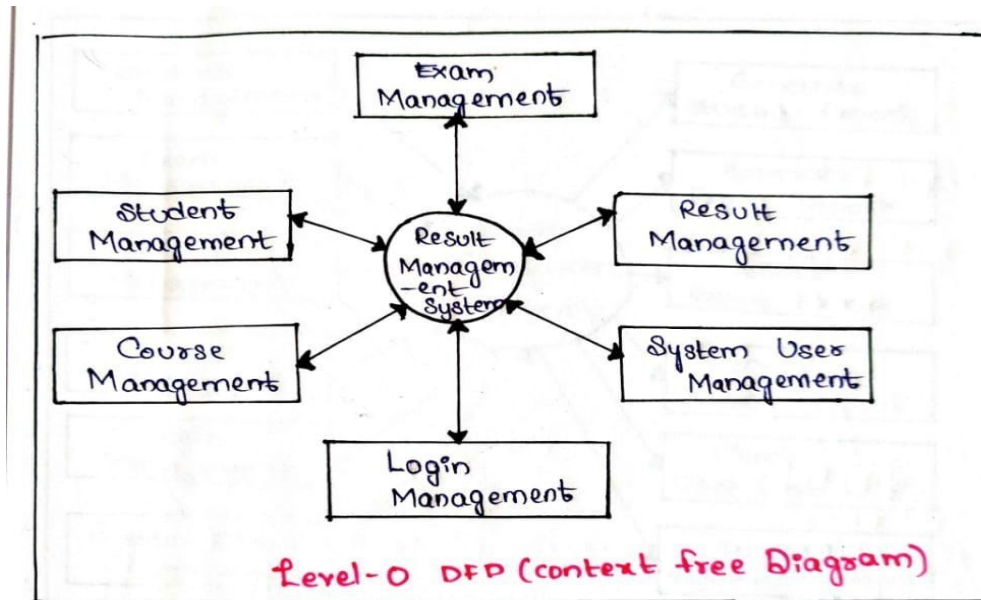
5.5 EFFICIENCY

It is efficient for all users because it easy to use and easy to understand. It has simple way of work that user wants to do.

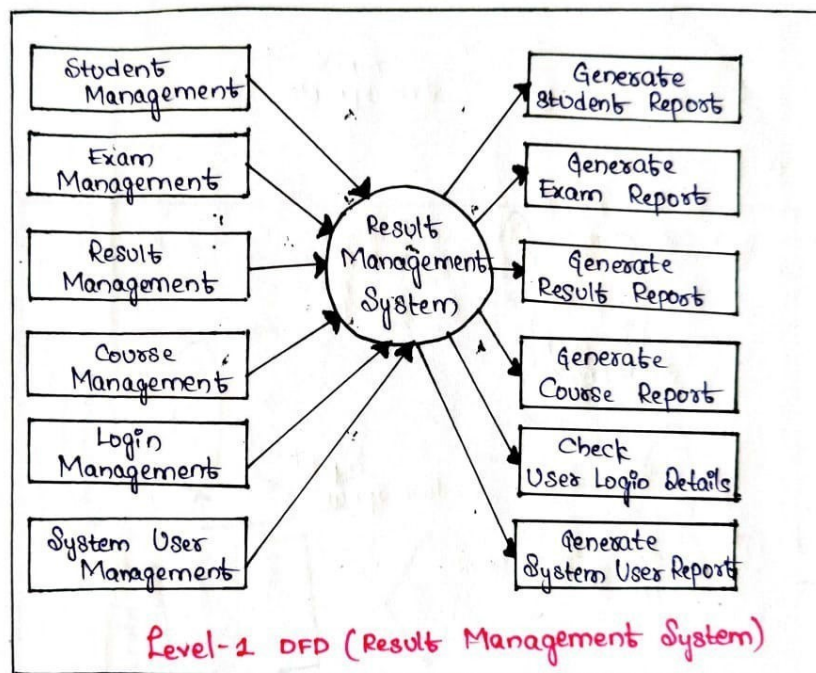
6. SYSTEM DESIGN

6.1 DFD Diagrams

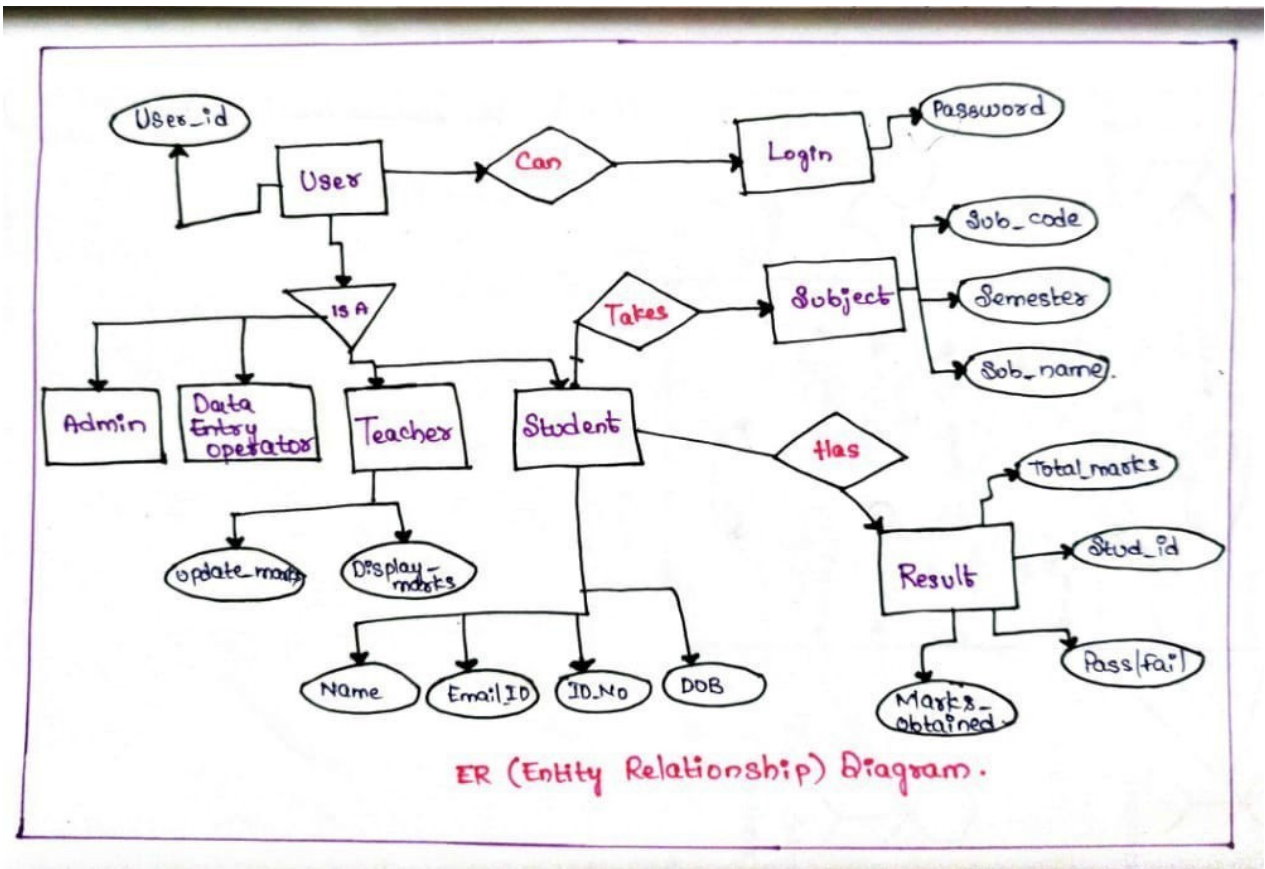
LEVEL ZERO :



LEVEL -1 Diagram

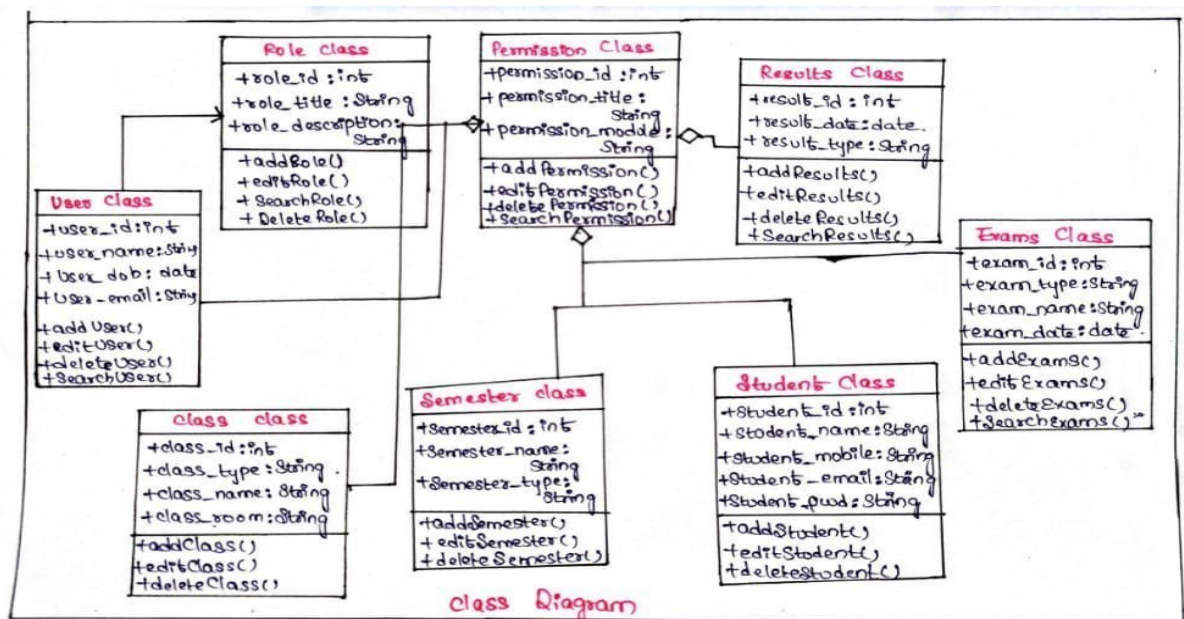


6.2 ER DIAGRAM

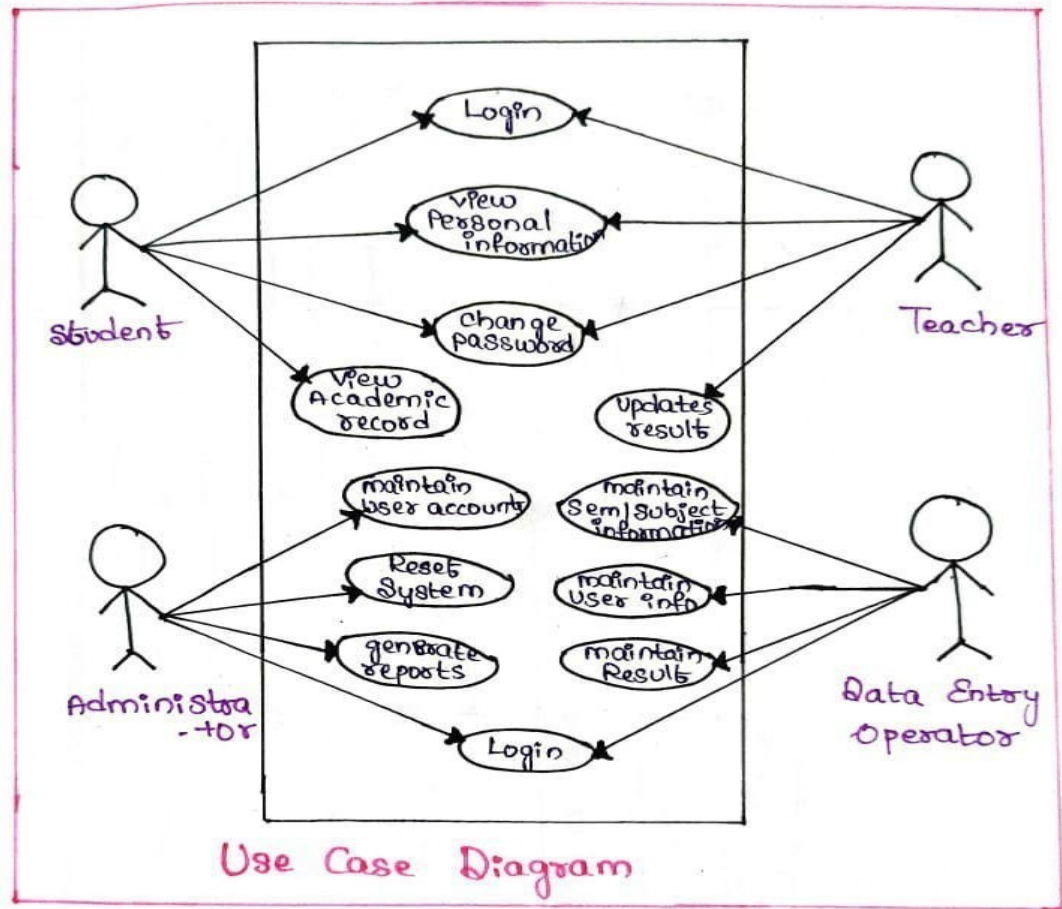


6.3 UML Diagrams

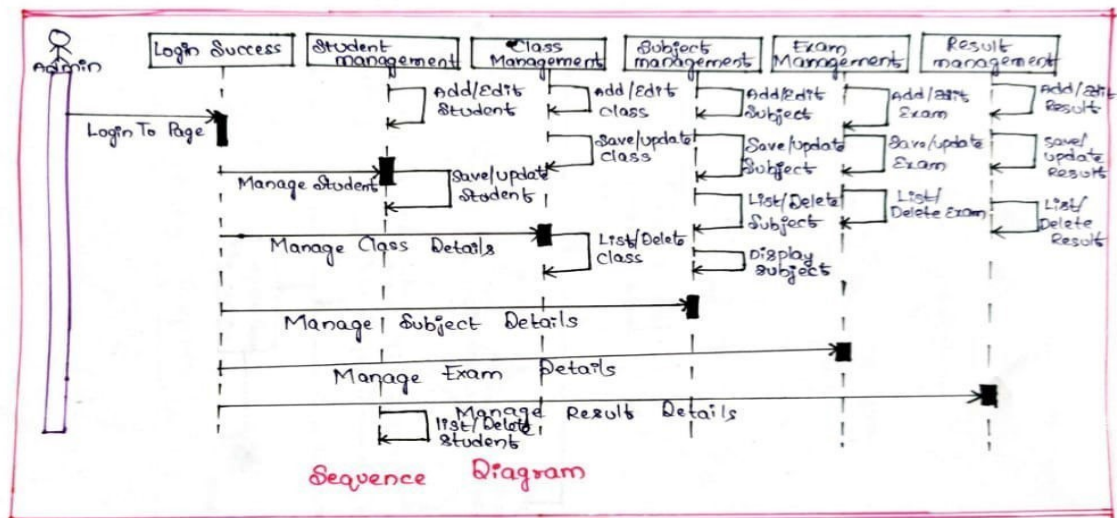
CLASS DIAGRAM



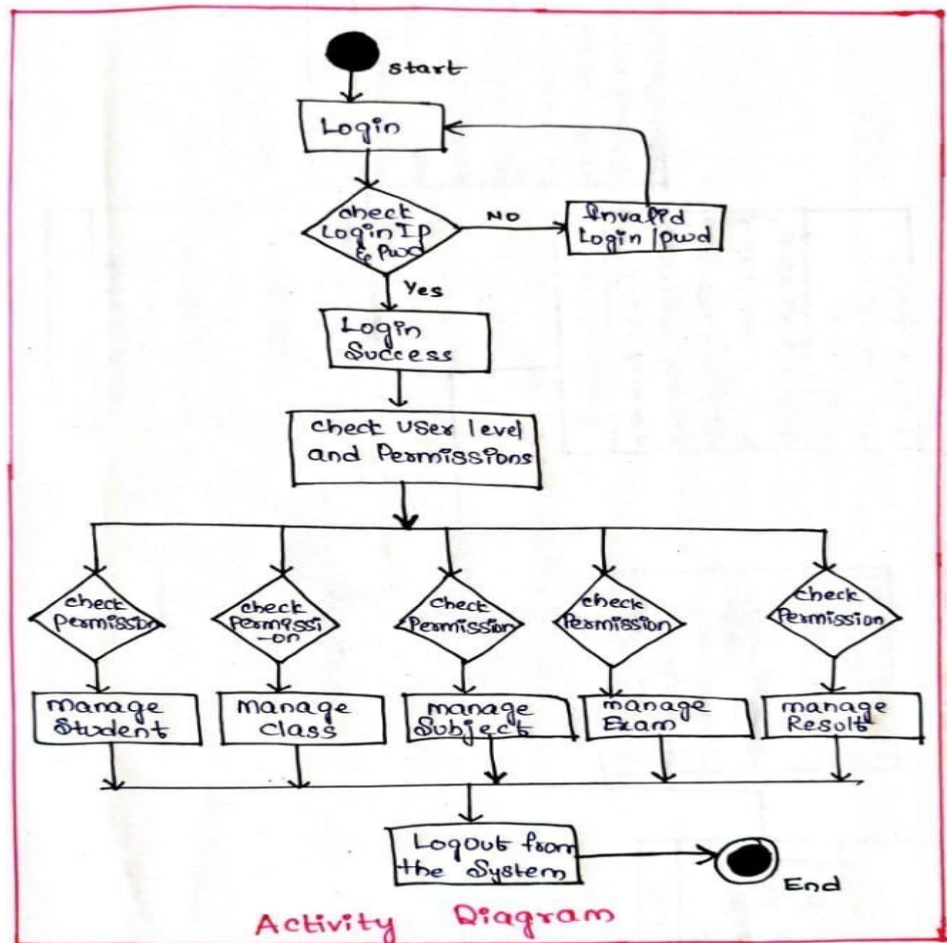
USECASE DIAGRAM



SEQUENCE DIAGRAM

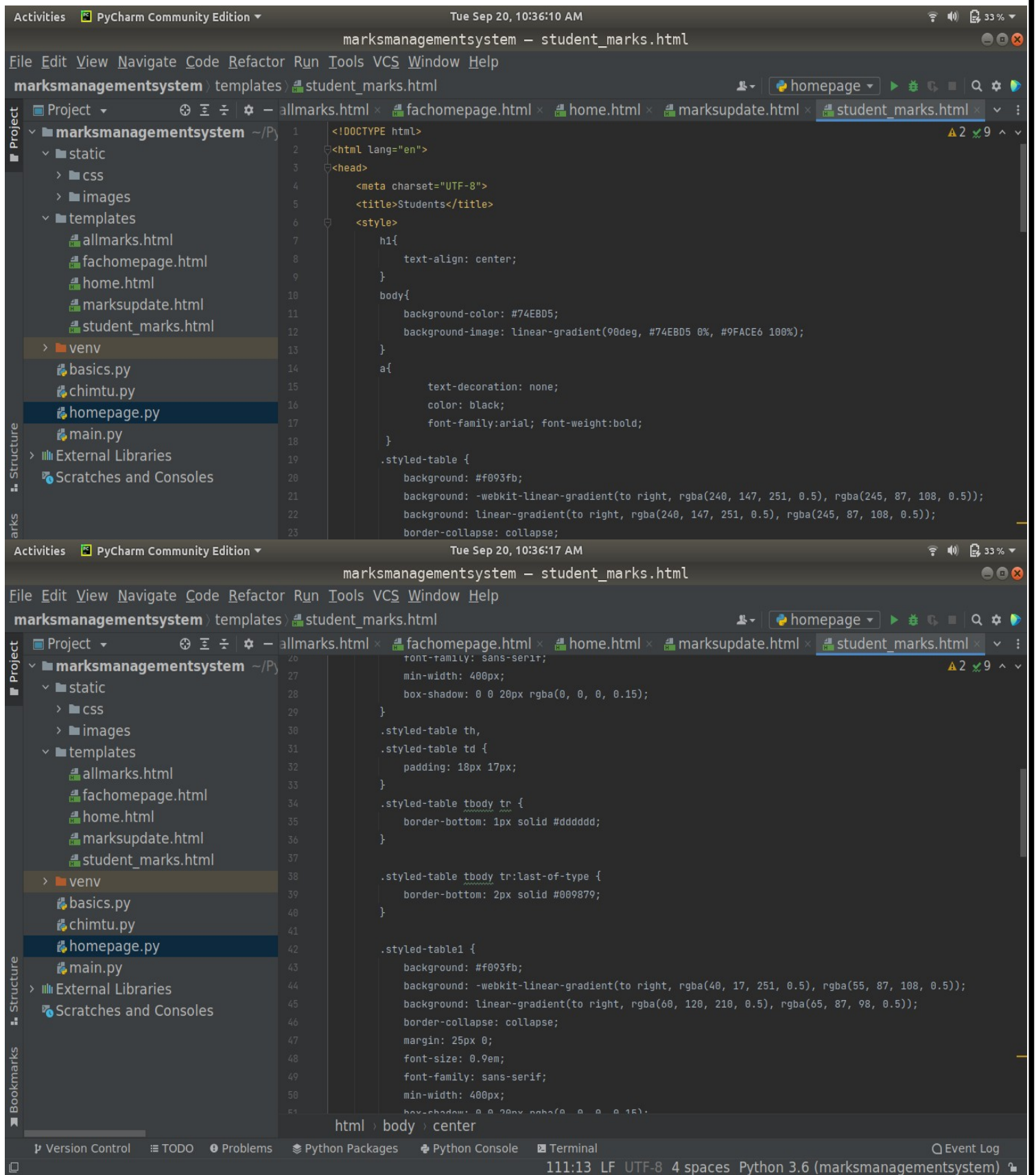


ACTIVITY DIAGRAM



7.Coding pics

7.1 Homepage.py



```

        if flag_marks == 1:
            print(i)
            if request.method == "post" and request.form["logout"] == "logout":
                return logout()
            return render_template("student_marks.html", res1=j, res2=flag_marks, res3=i)
        else:
            return render_template("student_marks.html", res1=j, res2=flag_marks, res3=i)
    else:
        if request.method == 'POST':
            result = request.form
            id = result.get("id")
            password = result.get("pswd")
            mycursor.execute("select * from students")
            temp = mycursor.fetchall()
            flag = 0
            i = ""
            for i in temp:
                if i[0] == id and i[1] == password:
                    flag = 1
                    break
            if (flag == 0):
                return fac_func(id, password)
            else:
                session["user"] = id

            mycursor.execute("select * from studentmarks")
            k = mycursor.fetchall()
            flag_marks = 0
            j = ""
            for j in k:
                if j[0] == id:
                    flag_marks = 1
                    break
            if flag_marks == 1:
                print(i)
                return render_template("student_marks.html", res1=j, res2=flag_marks, res3=i)
            else:
                return render_template("student_marks.html", res1=j, res2=flag_marks, res3=i)
        else:
            return student()

@app.route('/marksupdate', methods=['POST', 'GET'])
def marksupdate():
    mydb = mysql.connector.connect(
        host='localhost',
        user='root',
        password="",
        port='3306',
        database="test"
    )
    mycursor = mydb.cursor()

```

```

        database="test"
    )
    mycursor = mydb.cursor()
    mycursor.execute("select ID from teachers")
    a = mycursor.fetchall()
    teachers_id_list = []
    for i in a:
        teachers_id_list.append(i[0])
    print(teachers_id_list)
    if ("user" in session) and (session["user"] in teachers_id_list):
        mydb = mysql.connector.connect(
            host='localhost',
            user='root',
            password="",
            port='3306',
            database="test"
        )
        mycursor = mydb.cursor()
        if request.method == 'POST':
            result = request.form
            studid = result.get("sid")
            se = result.get("sesubject")
            cn = result.get("cnsubject")
            cd = result.get("cdsubject")
            os = result.get("ossubject")
            mycursor.execute("select ID from students where ID = '%s'" % (studid))
            k = mycursor.fetchall()
            print(k)
            if (se==" " or cn==" " or cd==" " or os==" " or int(se)>100 or int(cn)>100 or int(cd)>100 or int(os)>100):
                return render_template("marksupdate.html", result=1)
            if (len(k) != 0):
                mycursor.execute("insert into studentmarks(STUD_ID,SE,CN,CD,OS) values('%s','%s','%s','%s','%s')" % (
                    studid, se, cn, cd, os))
                mydb.commit()
                print("success")
                return render_template("marksupdate.html", result=0)
            else:
                print("failed")
                return render_template("marksupdate.html", result=1)
            return render_template("marksupdate.html", result=2)
        else:
            return student()

@app.route('/fachomepage', methods=['POST', 'GET'])
def fachomepage():
    mydb = mysql.connector.connect(
        host='localhost',
        user='root',
        password="",
        port='3306',

```

```

        database="test"
    )
    mycursor = mydb.cursor()
    mycursor.execute("select ID from teachers")
    a = mycursor.fetchall()
    teachers_id_list = []
    for i in a:
        teachers_id_list.append(i[0])
    print(teachers_id_list)
    if (("user" in session) and (session["user"] in teachers_id_list)):
        mycursor.execute("select PASSWORD from teachers where ID = '%s'" % (session["user"]))
        a = mycursor.fetchall()
        print(a)
        return fac_func(id=session["user"], password=a[0][0])
    return student()

```

```

def fac_func(id, password):
    mydb = mysql.connector.connect(
        host='localhost',
        user='root',
        password="",
        port='3306',
        database="test"
    )
    mycursor = mydb.cursor()
    mycursor.execute("select * from teachers")
    temp = mycursor.fetchall()
    flag_temp = 0
    i = ""
    for i in temp:
        if i[0] == id and i[1] == password:
            flag_temp = 1
            break
    if flag_temp == 1:
        session["user"] = id
        return render_template("fachomepage.html", result=i)
    else:
        return render_template("home.html", res=0)

```

```

@app.route('/allmarks')
def allmarks():
    mydb = mysql.connector.connect(
        host='localhost',
        user='root',
        password="",
        port='3306',
        database="test"
    )
    mycursor = mydb.cursor()

```

```

        database="test"
    )
    mycursor = mydb.cursor()
    mycursor.execute("select ID from teachers")
    a = mycursor.fetchall()
    teachers_id_list = []
    for i in a:
        teachers_id_list.append(i[0])
    print(teachers_id_list)
    if ("user" in session) and (session["user"] in teachers_id_list):
        mycursor.execute("select * from students")
        temp1 = mycursor.fetchall()
        mycursor.execute("select * from studentmarks")
        temp2 = mycursor.fetchall()
        l = []
        for i in temp2:
            k = []
            sid = i[0]
            k.append(sid)
            for j in temp1:
                if sid == j[0]:
                    k.append(j[2])
            k.extend(i[1:])
            l.append(k)
        print(l)
        return render_template("allmarks.html", result=l)
    else:
        return student()

```

```

@app.route('/logout')
def logout():
    session.pop("user", None)
    return student()

```

```

app.run(debug=True)

```


7.2 LOGINPAGE.html

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head><meta charset="UTF-8"><title>Marks System</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="{{url_for('static',filename='css/loginutil.css')}}">
<link rel="stylesheet" type="text/css" href="{{url_for('static',filename='css/loginmain.css')}}"></head>
<body>
<div class="limiter">
<div class="container-login100" style="background-image: url('{{url_for('static',filename = 'images/photo1.jpeg')}}');">
<div id="html-content-holder">
<div class="wrap-login100 p-l-55 p-r-55 p-t-65 p-b-54">
<form class="login100-form validate-form" action = "/home" method="POST">
<span class="login100-form-title p-b-49">Login</span>

<div class="wrap-input100 validate-input" data-validate="Userid is required">
<span class="label-input100">Username</span>
<input class="input100" type="text" name="id" placeholder="enter your userid">
<span class="focus-input100"></span>
</div>

<div class="wrap-input100 validate-input" data-validate = "Password is required">
<span class="label-input100">Password</span>
<input class="input100" type="password" name="pswd" placeholder="enter password">
<span class="focus-input100"></span>
</div>

<div class="text-right p-t-8 p-b-31">
<a href=" ">
Forgot password?
</a>
</div>

<div class="container-login100-form-btn">
<div class="wrap-login100-form-btn">
<div class="login100-form-bgbtn"></div>
<button name="submit_btn" class="login100-form-btn">
LOG IN
</button>
</div>
</div>
</form>
<br><br>
{%if res ==0 %}
<center><h4>INCORRECT ID OR PASSWORD</h4></center>
{%endif%}
</div>
</div>
</div>
</form>
</body>
</html>
```

7.3 STUDENTMARKS.html (when student logs in)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Students</title>
  <style>
    h1{
      text-align: center;
    }
    body{
      background-color: #74EBD5;
      background-image: linear-gradient(90deg, #74EBD5 0%, #9FACE6 100%);
    }
    a{
      text-decoration: none;
      color: black;
      font-family:arial; font-weight:bold;
    }
    .styled-table {
      background: #f093fb;
      background: -webkit-linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
      background: linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
      border-collapse: collapse;
      margin: 25px 0;
      font-size: 0.9em;
      font-family: sans-serif;
      min-width: 400px;
      box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);
    }
    .styled-table th,
    .styled-table td {
      padding: 18px 17px;
    }
    .styled-table tbody tr {
      border-bottom: 1px solid #dddddd;
    }
    .styled-table tbody tr:last-of-type {
      border-bottom: 2px solid #009879;
    }
    .styled-table1 {
      background: #f093fb;
      background: -webkit-linear-gradient(to right, rgba(40, 17, 251, 0.5), rgba(55, 87, 108, 0.5));
      background: linear-gradient(to right, rgba(60, 120, 210, 0.5), rgba(65, 87, 98, 0.5));
      border-collapse: collapse;
      margin: 25px 0;
      font-size: 0.9em;
      font-family: sans-serif;
      min-width: 400px;
      box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);
    }
  </style>

```

```

        font-family: sans-serif;
        min-width: 400px;
        box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);
    }

    .styled-table1 th,
    .styled-table1 td {
        padding: 18px 17px;
    }

    .styled-table1 tbody tr {
        border-bottom: 1px solid #dddddd;
    }

    .styled-table1 tbody tr:last-of-type {
        border-bottom: 2px solid #009879;
    }
</style>
</head>
<body style="background-image:url('{{url_for('static',filename = 'images/I1.jpg')}}');">
    <br><br>
    <h1 style="color:red; font-family:arial;">Student marks</h1>
    <br><br>
    <center>
    <table class="styled-table">
        <tr>
            <td>STUDENT ID :</td>
            <td>{{res3[0]}}</td>
        </tr>
        <tr>
            <td>STUDENT NAME :</td>
            <td>{{res3[2]}}</td>
        </tr>
        <tr>
            <td>STUDENT MAILID :</td>
            <td>{{res3[3]}}</td>
        </tr>
    </table>
    </center>
    <!--<h3>STUDENT ID :&nbsp;&nbsp;&nbsp;&{{res3[0]}}</h3>
    <h3>STUDENT NAME :&nbsp;&nbsp;&nbsp;&{{res3[2]}}</h3>
    <h3>STUDENT MAILID :&nbsp;&nbsp;&nbsp;&{{res3[3]}}</h3>-->
    <center>
    <table width="600px" border="2px" class="styled-table1">
        {% if res2 != 0 %}
        <tr>
            <th>STUDENT ID</th>
            <th>SE</th>
            <th>CN</th>
            <th>CD</th>
            <th>OS</th>
        </tr>
        <tr>
            {% for i in res1 %}

```

```
<th>CD</th>
<th>OS</th>
</tr>
<tr>
  {% for i in res1 %}
    <td>
      {{i}}
    </td>
    {% endfor %}
  </tr>
{% else %}
  <h3>MARKS NOT YET UPDATED FOR YOU!!!</h3>
{%endif %}
</table>
</center>
<center>
<form method="post">
<a href="/logout" name="logout"><h4>LOGOUT</h4></a>
</form>
</center>
</body>
</html>
```

7.4 FACULTYHOMEPAGE.html (when faculty logs in)

```
<!DOCTYPE HTML>

<html>
  <head>
    <title>HEAD PAGE</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
      .styled-table{
        border-radius: 10px;
        background-color: #74EBD5;
        background-image: linear-gradient(90deg, #74EBD5 0%, #9FACE6 100%);
      }
      a{
        text-decoration: none;
        color: white;
      }
      ::placeholder{
        color:red;
        opacity:1      ;
      }
      span{
        color:black;
      }
      body{
        background: #f093fb;
        background: -webkit-linear-gradient(to right, rgba(40, 17, 251, 0.5), rgba(55, 87, 108, 0.5));
        background: linear-gradient(to right, rgba(60, 120, 210, 0.5), rgba(65, 87, 98, 0.5));
      }
      .styled-table1{
        background: #f093fb;
        background: -webkit-linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
        background: linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
      }
      .styled-table1 th,
      .styled-table1 td {
        padding: 18px 17px;
      }
      .styled-table1 tbody tr {
        border-bottom: 1px solid #dddddd;
      }
      .styled-table1 tbody tr:last-of-type {
        border-bottom: 2px solid #009879;
      }
    </style>
  </head>
  <body style="background-image:url({{url_for('static',filename = 'images/form.jpg')}});">
    <form method="post">
      <table width = "100%" cellpadding="5px" frame = "box" bordercolor = "blue" class="styled-table"
        style="background-image:url({{url_for('static',filename = 'images/nav.jpg')}});">
        <tr>
          <td colspan="5" height="80px">
```

```

        <td colspan="5" height="80px">
            <h2 align="center" style="color:red; font-family:arial;">MARKS MANAGEMENT SYSTEM</h2>
        </td>
    </tr>
    <tr>
        <td align = "center">
            <a href = "/fahomepage" target = "_self" ><h3>HOME</h3></a>
        </td>
        <td align = "center">
            <a href = "/marksupdate" target = "_self" ><h3>UPDATE MARKS</h3></a>
        </td>
        <td align = "center">
            <a href = "/allmarks" target = "_self" ><h3>ALL STUDENTS</h3></a>
        </td>
        <td align = "center">
            <a href = "/logout" target = "_self" name = "submit" value = "logout"><h3>LOGOUT</h3></a>
        </td>
    </tr>
</table>
</form>
<center>
    <br><br><br>
<table width = "600px" frame = "box" bordercolor = "blue" class="styled-table1" style="border-radius:10px;">
    <tr>
        <td>TEACHER ID</td>
        <td>{{result[0]}}</td>
    </tr>
    <tr>
        <td>TEACHER NAME</td>
        <td>{{result[2]}}</td>
    </tr>
    <tr>
        <td>TEACHER MAIL ID</td>
        <td>{{result[3]}}</td>
    </tr>
</table>
</center>
</body>
</html>

```


7.5 MARKSUPDATEPAGE.html (marks update by faculty)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>MARKS UPDATE</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" href="{url_for('static',filename='css/util.css')}}">
  <link rel="stylesheet" type="text/css" href="{url_for('static',filename='css/main.css')}}">
  <style>
    .styled-table{
      border-radius: 10px;
      background-color: #74EBD5;
      background-image: linear-gradient(90deg, #74EBD5 0%, #9FACE6 100%);
    }
    a{
      text-decoration: none;
      color: white;
    }
    ::placeholder{
      color:red;
      opacity:1;
    }
    span{
      color:black;
    }
    body{
      background: #f093fb;
      background: -webkit-linear-gradient(to right, rgba(40, 17, 251, 0.5), rgba(55, 87, 108, 0.5));
      background: linear-gradient(to right, rgba(60, 120, 210, 0.5), rgba(65, 87, 98, 0.5));
    }
    .styled-table1{
      background: #f093fb;
      background: -webkit-linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
      background: linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
    }
    .styled-table1 th,
    .styled-table1 td {
      padding: 18px 17px;
    }
    .styled-table1 tbody tr {
      border-bottom: 1px solid #dddddd;
    }
    .styled-table1 tbody tr:last-of-type {
      border-bottom: 2px solid #009879;
    }
  </style>
</head>
<body style="background-image:url({url_for('static',filename = 'images/form.jpg')}});">
  <form method="post">
    <table width = "100%" cellpadding="5px" frame = "box" bordercolor = "blue" class="stvlved-table">
```

```

<form method="post">
<table width = "100%" cellpadding="5px" frame = "box" bordercolor = "blue" class="styled-table"
style="background-image:url({{url_for('static',filename = 'images/nav.jpg')}});">
<tr>
<td colspan="5" height="80px">
<h2 align="center" style="color:red; font-family:arial;">MARKS MANAGEMENT SYSTEM</h2>
</td>
</tr>
<tr>
<td align = "center">
<a href = "/fachomepage" target = "_self" ><h3>HOME</h3></a>
</td>
<td align = "center">
<a href = "/marksupdate" target = "_self" ><h3>UPDATE MARKS</h3></a>
</td>
<td align = "center">
<a href = "/allmarks" target = "_self" ><h3>ALL STUDENTS</h3></a>
</td>
<td align = "center">
<a href = "/logout" target = "_self" name = "submit" value = "logout"><h3>LOGOUT</h3></a>
</td>
</tr>
</table>
<div class="wrap-login100 p-l-50 p-r-50 p-t-72 p-b-50" style="margin-left: 33%; margin-top: 10px; border-radius: 25px;" >
{%if result==0 %}
<center><h3 style="color:pink;">SUCCESSFULLY REGISTERED STUDENT MARKS!!</h3></center>
{%elif result==1 %}
<center><h3 style="color:pink;">FAILED TO REGISTER STUDENT MARKS!!</h3></center>
{%endif%}<br><br>
<span class="login100-form-title p-b-59">
UPDATE MARKS
</span>
<div class="wrap-input100 validate-input" data-validate="Name is required">
<span class="label-input100">Student Id</span>
<input class="input100" type="text" name="sid" placeholder="Enter student id">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Name is required">
<span class="label-input100">SE MARKS</span>
<input class="input100" type="text" name="sesubject" placeholder="Enter SE marks">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Name is required">
<span class="label-input100">CN MARKS</span>
<input class="input100" type="text" name="cnsubject" placeholder="Enter CN marks">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Name is required">
<span class="label-input100">CD MARKS</span>
<input class="input100" type="text" name="cdsubject" placeholder="Enter CD marks">
<span class="focus-input100"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Name is required">

```



```

<div class="wrap-input100 validate-input" data-validate="Name is required">
  <span class="label-input100">CD MARKS</span>
  <input class="input100" type="text" name="cdsubject" placeholder="Enter CD marks">
  <span class="focus-input100"></span>
</div>

<div class="wrap-input100 validate-input" data-validate="Name is required">
  <span class="label-input100">OS MARKS</span>
  <input class="input100" type="text" name="ossubject" placeholder="Enter OS MARKS">
  <span class="focus-input100"></span>
</div>

<div class="container-login100-form-btn">
  <div class="wrap-login100-form-btn">
    <div class="login100-form-bggbtn"></div>
    <button name="submit" class="login100-form-btn" style="align-self:center;">
      Submit marks
    </button>
  </div>
</div>
</div>
</form>
</body>
</html>

```

7.6 ALLSTUDENTMARKS.html (viewed for faculty)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Students all marks</title>
  <style>
    h1{
      background-color: powderblue;
      text-align: center;
    }
    .styled-table{
      border-radius: 10px;
      background-color: #74EBD5;
      background-image: linear-gradient(90deg, #74EBD5 0%, #9FACE6 100%);
    }
    a{
      text-decoration: none;
      color: white;
    }
    ::placeholder{
      color:red;
      opacity:1;
    }
    span{
      color:black;
    }
    body{
      background: #f093fb;
      background: -webkit-linear-gradient(to right, rgba(40, 17, 251, 0.5), rgba(55, 87, 108, 0.5));
      background: linear-gradient(to right, rgba(60, 120, 210, 0.5), rgba(65, 87, 98, 0.5));
    }
    .styled-table1{
      background: #f093fb;
      background: -webkit-linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
      background: linear-gradient(to right, rgba(240, 147, 251, 0.5), rgba(245, 87, 108, 0.5));
    }
    .styled-table1 th,
    .styled-table1 td {
      padding: 18px 17px;
    }
    .styled-table1 tbody tr {
      border-bottom: 1px solid #dddddd;
    }
    .styled-table1 tbody tr:last-of-type {
      border-bottom: 2px solid #009879;
    }
  </style>
</head>
<body style="background-image:url({{url_for('static',filename = 'images/form.jpg')}});">
  <form method="post">
```

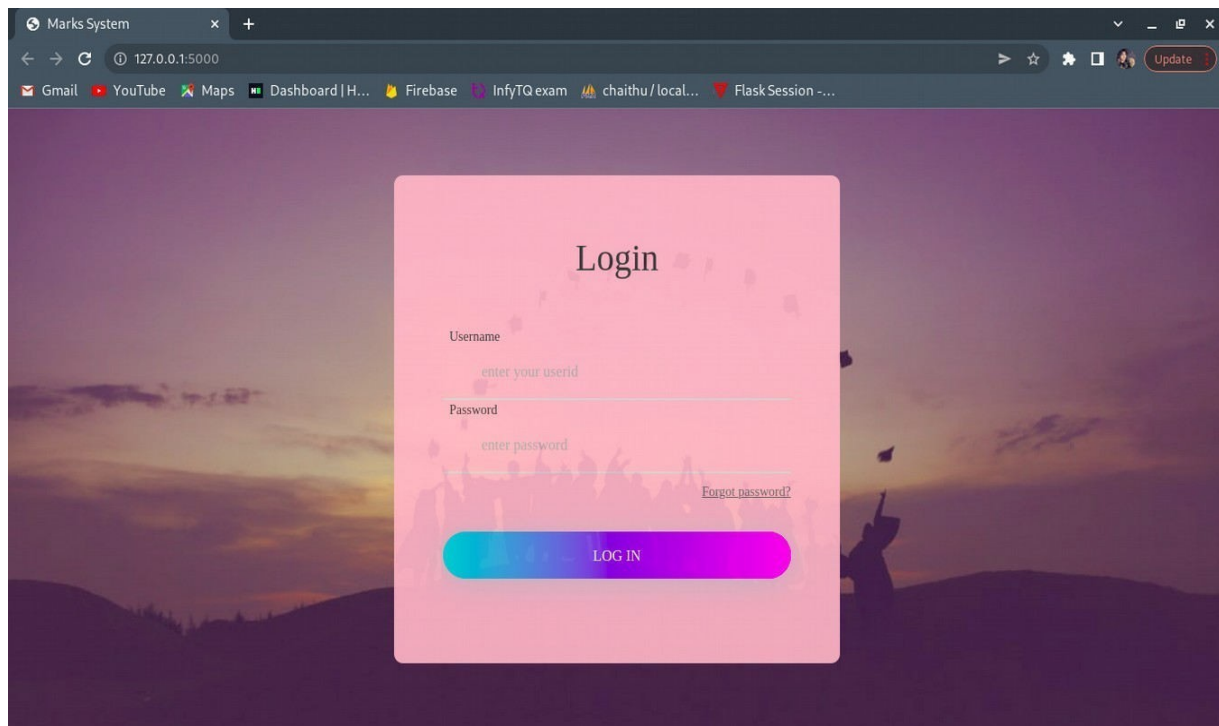
```

<body style="background-image:url({{url_for('static',filename = 'images/form.jpg')}});">
    <form method="post">
        <table width = "100%" cellpadding="5px" frame = "box" bordercolor = "blue"
            class="styled-table" style="background-image:url({{url_for('static',filename = 'images/nav.jpg')}});">
            <tr>
                <td colspan="5" height="80px">
                    <h2 align="center" style="color:red; font-family:arial;">MARKS MANAGEMENT SYSTEM</h2>
                </td>
            </tr>
            <tr>
                <td align = "center">
                    <a href = "/fachomepage" target = "_self" ><h3>HOME</h3></a>
                </td>
                <td align = "center">
                    <a href = "/marksupdate" target = "_self" ><h3>UPDATE MARKS</h3></a>
                </td>
                <td align = "center">
                    <a href = "/allmarks" target = "_self" ><h3>ALL STUDENTS</h3></a>
                </td>
                <td align = "center">
                    <a href = "/logout" target = "_self" name = "submit" value = "logout"><h3>LOGOUT</h3></a>
                </td>
            </tr>
        </table>
    </form>
    <center>
        <h2 style="color:green; font-family:arial;">ALL STUDENT MARKS</h2>
        <table width = "600px" cellpadding="5px" frame = "box" bordercolor = "blue" class="styled-table1">
            <tr>
                <th>STUDENT ID</th>
                <th>STUDENT NAME</th>
                <th>SE</th>
                <th>CN</th>
                <th>CD</th>
                <th>OS</th>
            </tr>
            {%for i in result%}
            <tr>
                {% for j in i %}
                <td>
                    {{j}}
                </td>
                {% endfor %}
            </tr>
            {%endfor %}
        </table>
    </center>
</body>
</html>

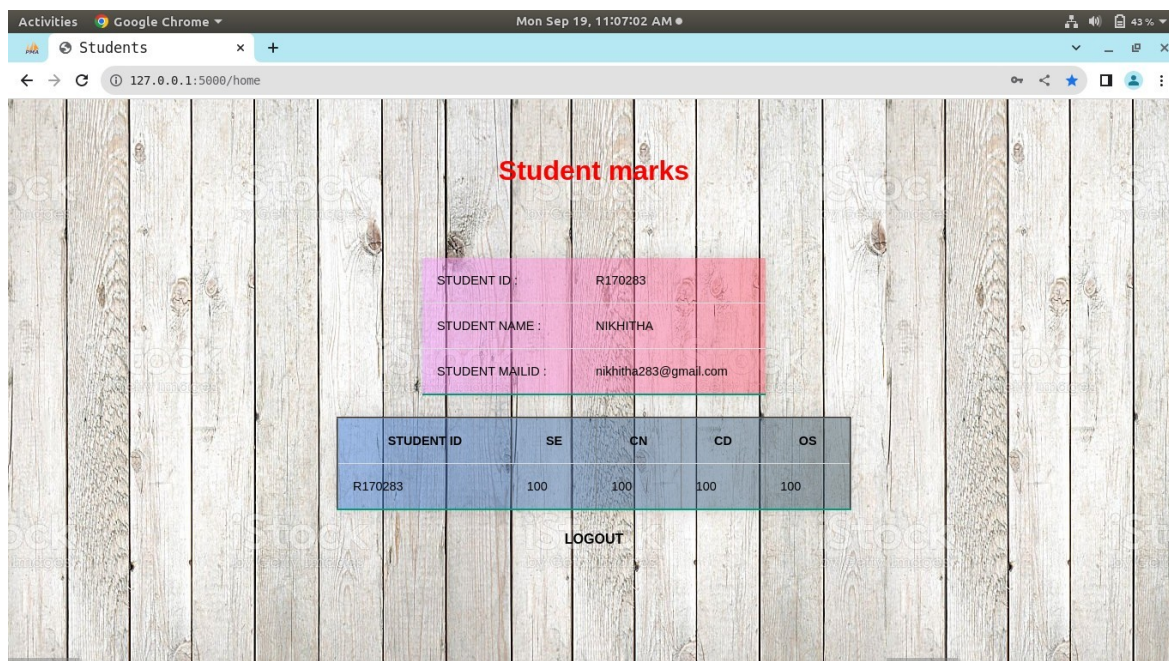
```

8. Output

8.1 Login page



8.2 Student marks page (When student logs in)



8.3 Faculty Home Page (When faculty logs in)

The screenshot shows a web browser window with the title 'HEAD PAGE' and the URL '127.0.0.1:5000/home'. The page features a blue header with the text 'MARKS MANAGEMENT SYSTEM' and four navigation links: 'HOME', 'Enter MARKS', 'ALL STUDENTS', and 'LOGOUT'. Below the header, there is a pink box containing the following information:

TEACHER ID	tester
TEACHER NAME	VINOD
TEACHER MAIL ID	vinod@gmail.com

8.4 Enter marks page (In Faculty page)

The screenshot shows a web browser window with the title 'MARKS UPDATE' and the URL '127.0.0.1:5000/markupdate'. The page features a blue header with the text 'MARKS MANAGEMENT SYSTEM' and four navigation links: 'HOME', 'Enter MARKS', 'ALL STUDENTS', and 'LOGOUT'. Below the header, there is a white box with the title 'Enter MARKS' and the following form fields:

Student Id
Enter student id

SE MARKS
Enter SE marks

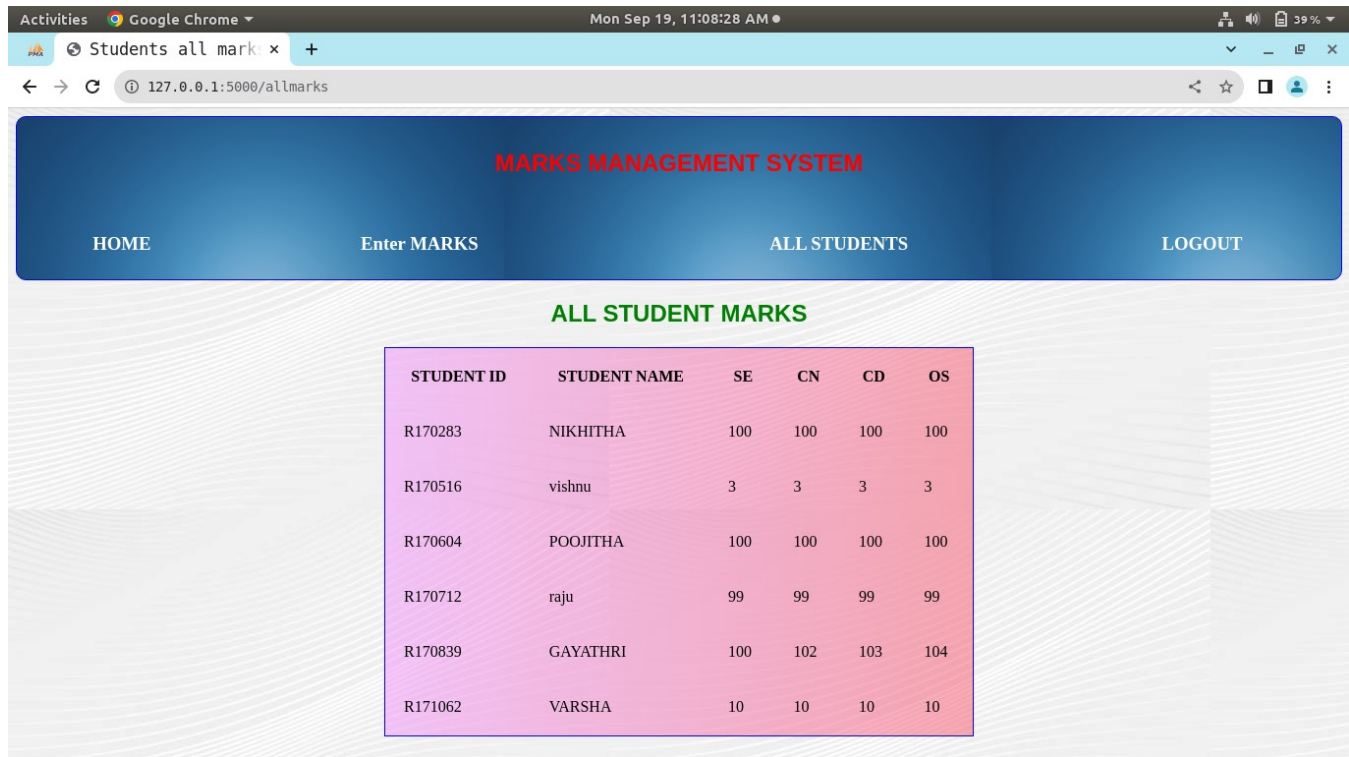
CN MARKS
Enter CN marks

CD MARKS
Enter CD marks

OS MARKS
Enter OS MARKS

Submit marks

8.5 All student marks Page (In Faculty page)



The screenshot shows a web browser window with the title 'Students all marks'. The address bar shows the URL '127.0.0.1:5000/allmarks'. The page has a blue header with the title 'MARKS MANAGEMENT SYSTEM' and four navigation buttons: 'HOME', 'Enter MARKS', 'ALL STUDENTS', and 'LOGOUT'. Below the header, the page title 'ALL STUDENT MARKS' is displayed in green. A table with six columns (STUDENT ID, STUDENT NAME, SE, CN, CD, OS) and seven rows of student data is shown.

STUDENT ID	STUDENT NAME	SE	CN	CD	OS
R170283	NIKHITHA	100	100	100	100
R170516	vishnu	3	3	3	3
R170604	POOJITHA	100	100	100	100
R170712	raju	99	99	99	99
R170839	GAYATHRI	100	102	103	104
R171062	VARSHA	10	10	10	10

References:

<https://www.w3schools.com/php/default.asp>

<https://www.mysql.com/>

<https://www.apachefriends.org/download.html>

*****THANK YOU*****