

BIBLIOGRAPHY

Appendix-A

URL Listings

Websites	Data collected
https://wikipedia.org	Searching of any information that will be used in documentation.
https://dev.sqlserver.com/doc	SQL server it performing in mainly depending on the one of the database using.
https://www.answers.com	Answers.com, online dictionary, encyclopedia and much more.
https://google.co.in	Any information searching and downloading.
https://training-classes.com	Designing part information as gathered

References

- [1] Bhattacharya, S., Nainala, G. S., Das, P., & Routray, A. (2018, July). Smart attendance monitoring system (SAMS): a face recognition based attendance system for classroom environment. In 2018 , 8th International Conference on Advanced Learning Technologies (ICALT) (pp. 358-360).
- [2] Gagare, P. S., Sathe, P. A., Pawaskar, V. T., & Bhawe, S. S. (2014). Smart attendance system. International Journal on Recent and Innovation Trends in Computing and Communication, 2(1), 124-127.
- [3] Patel, R., Patel, N., & Gajjar, M. (2012). Online students' attendance monitoring system in classroom using radio frequency identification technology: a proposed system framework. International Journal of Emerging Technology and Advanced Engineering, 2(2), 61-66.
- [4] Mothwa, L., Tapamo, J. R., & Mapati, T. (2018, November). Conceptual model of the smart attendance monitoring system using computer vision. In 2018 14th International Conference on SignalImage Technology & Internet-Based Systems (SITIS) (pp. 229-234).

- [5] Shoewu, O., & Idowu, O. A. (2012). Development of attendance management system using biometrics. *The Pacific Journal of Science and Technology*, 13(1), 300-307.
- [6] Rahni, A. A., Zainal, N., Adna, M. Z., Othman, N. E., & Bukhori, M. F. (2015). Development of the online student attendance monitoring system (SAMSTM) based on QR-codes and mobile devices. *J. Eng. Sci. Technol*, 10, 28-40.
- [7] Sawhney, S., Kacker, K., Jain, S., Singh, S. N., & Garg, R. (2019, January). Real-time smart attendance system using face recognition techniques. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 522-525).
- [8] Patel, R., Patel, N., & Gajjar, M. (2012). Online students' attendance monitoring system in classroom using radio frequency identification technology: a proposed system framework. *International Journal of Emerging Technology and Advanced Engineering*, 2(2), 61-66.
- [9] Wei, X., Manori, A., Devnath, N., Pasi, N., & Kumar, V. (2018). QR Code Based Smart Attendance System. no. October.
- [10] Chintalapati, S., & Raghunadh, M. V. (2013, December). Automated attendance management system based on face recognition algorithms. In *2013, International Conference on Computational Intelligence and Computing Research* (pp. 1-5).

Appendix – B**Glossary**

API	→	Application Programming Interface
GUI	→	Graphical User Interface
HTML	→	Hyper Text Markup Language
HTTP	→	Hyper Text Transfer Protocol
SQL	→	Structured Query Language
UML	→	Unified Modeling Language
URL	→	Uniform Resource Locator
WWW	→	World Wide Web
SRS	→	Software Requirement Specification
RFID	→	Radio Frequency Identification Detection
UPC	→	Universal Product Code
QRC	→	Quick Response Code
NFC	→	Near Field Communication
MAC	→	Media Access Control
LMS	→	Learning Management System

Appendix-C

List of Tables

SNo	Table No	Title of Table	Page No	Chapter
1	3.2.1	Faculty	26	System Design
2	3.2.2	Student	26	System Design
3	4.1.1	Black Box Testing	49	Testing
4	4.1.2	White Box Testing	50	Testing
5	4.2.1	Test Cases for Overall system	56	Testing

List of Figures

SNo	Figure No	Title of Figure	Page No	Chapter
1	2.1.1	System Architecture	9	SRS
2	2.4.1	Triple Constraints of Project Management	11	SRS
3	2.7.1	Non-functional Requirments	13	SRS
4	2.7.2	Analysis Model	17	SRS
5	2.7.3	Designing Stage	18	SRS
6	2.7.4	Development Stage	19	SRS

7	2.7.5	Integration & Test Stage	20	SRS
8	2.7.6	Spiral Model	22	SRS
9	3.1.1	ER Diagram for Overall System	25	System Design
10	3.3.1	Use Case Diagram for Overall System	37	System Design
11	3.3.2	Class Diagram for Overall System	38	System Design
12	3.3.3	Sequence Diagram for Faculty	40	System Design
13	3.3.4	Sequence Diagram for Student	41	System Design
14	3.3.5	Activity Diagram for Faculty	43	System Design
15	3.3.6	Activity Diagram for Student	44	System Design
16	3.3.7	Deployment Diagram for Overall System	45	System Design
17	4.1	Levels of Testing	46	Testing
18	4.2	Testing Methodologies	47	Testing

List of Screens

SNo	Screen No	Title of Screen	Page No	Chapter
1	5.1.1	Home page	67	Implementation
2	5.1.2	Faculty Login page	68	Implementation
3	5.1.3	Faculty Home page	69	Implementation
4	5.1.4	Create student data page	70	Implementation
5	5.1.5	Generate QR Code page	71	Implementation
6	5.1.6	QR Code Image page	72	Implementation
7	5.1.7	QR Code Scanning page	73	Implementation
8	5.1.8	View student data page	74	Implementation
9	5.1.9	View student Attendance Login page	75	Implementation
10	5.1.10	View student Attendance Home page	76	Implementation
11	5.1.11	Faculty Logout page	77	Implementation
12	5.1.12	Student Login page	78	Implementation
13	5.1.13	Student Home page	79	Implementation
14	5.1.14	Student Check Attendance Login page	80	Implementation
15	5.1.15	Student Check Attendance Home page	81	Implementation
16	5.1.16	Student Logout page	82	Implementation

Appendix- D

Coding

```
import os
from django.core.files.storage import FileSystemStorage
import pymysql
import datetime
import pyqrcode
import png
from pyqrcode import QRCode
from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
from django.http import HttpResponseRedirect
global username

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

def FacultyLoginAction(request):
    global username
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if username == 'faculty' and password == 'faculty':
            context= {'data':'Hello! Faculty Member'}
            return render(request, 'FacultyScreen.html', context)
        else:
            context= {'data':'login failed. Please retry'}
            return render(request, 'FacultyLogin.html', context)

def FacultyLogin(request):
    if request.method == 'GET':
```

```

        return render(request, 'FacultyLogin.html', {})

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

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

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

def ViewStudAttendanceAction(request):
    if request.method == 'POST':
        studid = request.POST.get('t1', False)
        from_date = request.POST.get('t2', False)
        to_date = request.POST.get('t3', False)
        from_dd = str(datetime.datetime.strptime(from_date, "%d-%b-%Y").strftime("%Y-%m-%d"))
        to_dd = str(datetime.datetime.strptime(to_date, "%d-%b-%Y").strftime("%Y-%m-%d"))
        presence_days = 0
        Attendance = 0
        columns = ['Student ID', 'Presence Date']
        output = '<table border=1 align=center width=100%>'
        font = '<font size="" color="black">'
        output += "<tr>"
        for i in range(len(columns)):
            output += "<th>" + font + columns[i] + "</th>"
        output += "</tr>"
        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'root', database = 'stud_attendance',charset='utf8')
        with con:

```



```

        cur = con.cursor()
        cur.execute("select stud_attendance FROM stud_details where
studentID='"+studid+"'")
        rows = cur.fetchall()
        for row in rows:
            attendance = row[0]
            break
        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'root', database = 'stud_attendance',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * from mark_attendance where
studentID='"+studid+"' and attended_date between '"+from_dd+"' and
 '"+to_dd)
            rows = cur.fetchall()
            for row in rows:
                presence_days = presence_days + 1
                output += "<tr>"
                output += "<td>"+font+str(row[0])+"</td>"
                output += "<td>"+font+str(row[1])+"</td></tr>"
                output+="<tr><td>"+font+"AttendedDays:
"+str(presence_days)+"</font><td>"+font+"Attendance="+str(((attendance
/30)* presence_days))+"</td></tr>"
            context= {'data': output}
            return render(request, 'FacultyScreen.html', context)

def ViewStudAttendance(request):
    if request.method == 'GET':
        font = '<font size="" color="black">'
        output = '<tr><td>'+font+'Choose&nbsp;Stud ID</td><td><select
name="t1">'
        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'root', database = 'stud_attendance',charset='utf8')
        with con:

```

```

cur = con.cursor()
cur.execute("select studentID FROM student_details")
rows = cur.fetchall()
for row in rows:
    output += '<option value="'+row[0]+'">'+row[0]+'</option>'
output += "</select></td></tr>"
context= {'data1': output}
return render(request, 'ViewStudAttendance.html', context)

```

```
def ViewAttendance(request):
```

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

```
        return render(request, 'ViewAttendance.html', { })
```

```
def ViewAttendanceAction(request):
```

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

```
        global username
```

```
        studid = username
```

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

```
        to_date = request.POST.get('t2', False)
```

```
        from_dd = str(datetime.datetime.strptime(from_date, "%d-%b-%Y").strftime("%Y-%m-%d"))
```

```
        to_dd = str(datetime.datetime.strptime(to_date, "%d-%b-%Y").strftime("%Y-%m-%d"))
```

```
        presence_days = 0
```

```
        attendance = 0
```

```
        columns = ['Student ID', 'Attended Date']
```

```
        output = '<table border=1 align=center width=100%>'
```

```
        font = '<font size="" color="black">'
```

```
        output += "<tr>"
```

```
        for i in range(len(columns)):
```

```
            output += "<th>"+font+columns[i]+"</th>"
```

```
        output += "</tr>"
```

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

```

with con:
    cur = con.cursor()

cur.execute("select stud_attendance FROM student_details where
StudentID='"+empid+"'")
    rows = cur.fetchall()
    for row in rows:
        Attendance = row[0]
        break
    con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'root', database = 'stud_attendance',charset='utf8')
with con:
    cur = con.cursor()
        cur.execute("select * from mark_attendance where
studentID='"+empid+"' and attended_date between '"+from_dd+"' and
"+to_dd)
    rows = cur.fetchall()
    for row in rows:
        presence_days = presence_days + 1
        output += "<tr>"
        output += "<td>"+font+str(row[0])+"</td>"
        output += "<td>"+font+str(row[1])+"</td></tr>"
        output+="<tr><td>"+font+"AttendedDays:
"+str(presence_days)+"</font><td>"+font+"Attendance="+str(((Attendance/30)* presence_days))+"</td></tr>"
    context= {'data': output}
    return render(request, 'UserScreen.html', context)

def ViewStud(request):
    if request.method == 'GET':
        columns = ['Stud ID', 'Name', 'Phone No', 'Course', 'Attendance']
        output = '<table border=1 align=center width=100%>'
        font = '<font size="" color="black">'
        output += "<tr>"

```

```
for i in range(len(columns)):
    output += "<th>" + font + columns[i] + "</th>"
output += "</tr>"

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

with con:
    cur = con.cursor()
    cur.execute("select * FROM student_details")
    rows = cur.fetchall()
    for row in rows:
        output += "<tr>"
        output += "<td>" + font + str(row[0]) + "</td>"
        output += "<td>" + font + str(row[1]) + "</td>"
        output += "<td>" + font + str(row[2]) + "</td>"
        output += "<td>" + font + str(row[3]) + "</td>"
        output += "<td>" + font + str(row[4]) + "</td></tr>"

context= {'data': output}
return render(request, 'FacultyScreen.html', context)
```

```
def FacultyLoginAction(request):
    global username
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        index = 0
        stud_name = None

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

        with con:
            cur = con.cursor()
            cur.execute("select studentID, studentName FROM student_details")
            rows = cur.fetchall()
            for row in rows:
                if row[0] == username:
                    stud_name = row[1]
```

```
        index = 1
        break
    if index == 1:
        context= {'data': 'welcome '+stud_name}
        return render(request, 'UserScreen.html', context)
    else:
        context= {'data': 'login failed. Please retry'}
        return render(request, 'UserLogin.html', context)

def DownloadAction(request):
    if request.method == 'POST':
        global username
        infile = open("StudentAttendance/static/qrcodes/"+username+".png",
'rb')
        data = infile.read()
        infile.close()

        response = HttpResponse(data, content_type='image/png')
        response['Content-Disposition'] = 'attachment; filename=%s' %
username+".png"
        return response

def AddStudAction(request):
    if request.method == 'POST':
        global username
        ids = request.POST.get('t1', False)
        name = request.POST.get('t2', False)
        phone = request.POST.get('t3', False)
        desg = request.POST.get('t4', False)
        sal = request.POST.get('t5', False)
        output = "none"

        con = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root',
password = 'root', database = 'stud_attendance',charset='utf8')
        with con:
```

```
cur = con.cursor()
cur.execute("select studentID FROM student_details")
rows = cur.fetchall()
for row in rows:
    if row[0] == studid:
        output = ids+" student already exists"
        break
if output == 'none':
    db_connection = pymysql.connect(host='127.0.0.1',port = 3306,user
= 'root', password = 'root', database = 'stud_attendance',charset='utf8')
    db_cursor = db_connection.cursor()
    student_sql_query="INSERT INTO
student_details(studentID,studentName,phoneNo,course,Attendance)
VALUES('"+ids+"','"+name+"','"+phone+"','"+course+"','"+attendance+"')"
    db_cursor.execute(student_sql_query)
    db_connection.commit()
    url = pyqrcode.create(ids)
    url.png('StudentAttendance/static/qrcodes/'+ids+'.png', scale = 6)
    username = ids
    print(db_cursor.rowcount, "Record Inserted")
    if db_cursor.rowcount == 1:
        output = 'Stud Details Saved with ID : '+ids
    context= {'data':output}
    return render(request, 'Download.html', context)
```