**QR CODE BASED ATTENDANCE SYSTEM**

**1.REQUIRMENTS WORKFLOW**

**USECASE DIAGRAM :**

****

**USECASE DESCRIPTION :**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioraldiagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

**2.ANALYSIS WORKFLOW :**

**2.1 OBJECT ORIENT ANALYSIS :**

Object-Oriented Analysis (OOA): Object-Oriented Analysis (OOA) is the first technical activity performed as part of object-oriented software engineering. OOA introduces new concepts to investigate a problem. It is based on a set of basic principles, which are as follows-

1.The information domain is modeled.

2.Behavior is represented.

3.The function is described.

4. Data, functional, and behavioral models are divided to uncover greater detail.

5.Early models represent the essence of the problem, while later ones provide implementation details.

**CLASS DIAGRAM :**

****

**SEQUENCE DIAGRAM :**

****

**3.DESIGN WORKFLOW :**

**3.1 DATA FLOW ANALYSIS**

It is the analysis of flow of data in control flow graph, i.e., the analysis that determines the information regarding the definition and use of data in program. With the help of this analysis, optimization can be done. In general, its process in which values are computed using data flow analysis. The data flow property represents information that can be used for optimization.

**Basic Terminologies –**

Definition Point: a point in a program containing some definition.

Reference Point: a point in a program containing a reference to a data item.

Evaluation Point: a point in a program containing evaluation of expression.

Lightbox

**Advantage –**

It is used to eliminate common sub expressions.

**MODULE DIAGRAM**

**MODULE DESCRIPTION**

1. Admin module: admin can login to application by using username and password as ‘admin’ and ‘admin’ and then can ADD New Employee Details and then application will generate QR CODE on EMPLOYEE ID and then admin can download that image and give to employee and employee can show that image to QR CODE scanner to mark attendance. Admin can view all employee details and then can view employee attendance by using start and end date.
2. Employee Login Module: employee can login to system by using his ID and can view his attendance from start and end date selection
3. QR CODE WEBCAM SCANNER: employee has to show his QR CODE image from his mobile to webcam and then webcam will read QR CODE and mark attendance. Only one attendance for each employee for each day will be marked.

**PSEUDOCODE**

**4.IMPLEMENTED WORKFLOW :**

**SOURCE CODE :**

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 HttpResponse

global username

def test(request):

if request.method == 'GET':

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

def AdminLoginAction(request):

global username

if request.method == 'POST':

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

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

if username == 'admin' and password == 'admin':

context= {'data':'Hello! Administrator'}

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

else:

context= {'data':'login failed. Please retry'}

return render(request, 'AdminLogin.html', context)

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 index(request):

if request.method == 'GET':

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

def AddEmp(request):

if request.method == 'GET':

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

def ViewEmpAttendanceAction(request):

if request.method == 'POST':

empid = 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

salary = 0

columns = ['Employee 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 = 'emp\_attendance',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select emp\_salary FROM employee\_details where employeeID='"+empid+"'")

rows = cur.fetchall()

for row in rows:

salary = row[0]

break

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

with con:

cur = con.cursor()

cur.execute("select \* from mark\_attendance where employeeID='"+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+"Attended Days : "+str(presence\_days)+"</font><td>"+font+"Current Salary = "+str(((salary/30) \* presence\_days))+"</td></tr>"

context= {'data': output}

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

def ViewEmpAttendance(request):

if request.method == 'GET':

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

output = '<tr><td>'+font+'Choose&nbsp;Emp ID</td><td><select name="t1">'

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

with con:

cur = con.cursor()

cur.execute("select employeeID FROM employee\_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, 'ViewEmpAttendance.html', context)

def ViewAttendance(request):

if request.method == 'GET':

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

def ViewAttendanceAction(request):

if request.method == 'POST':

global username

empid = 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

salary = 0

columns = ['Emp 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 = 'emp\_attendance',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select emp\_salary FROM employee\_details where employeeID='"+empid+"'")

rows = cur.fetchall()

for row in rows:

salary = row[0]

break

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

with con:

cur = con.cursor()

cur.execute("select \* from mark\_attendance where employeeID='"+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+"Attended Days : "+str(presence\_days)+"</font><td>"+font+"Current Salary = "+str(((salary/30) \* presence\_days))+"</td></tr>"

context= {'data': output}

return render(request, 'UserScreen.html', context)

def ViewEmp(request):

if request.method == 'GET':

columns = ['Emp ID', 'Name', 'Phone No', 'Designation', 'Salary']

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 = 'emp\_attendance',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select \* FROM employee\_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, 'AdminScreen.html', context)

def UserLoginAction(request):

global username

if request.method == 'POST':

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

index = 0

emp\_name = None

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

with con:

cur = con.cursor()

cur.execute("select employeeID, empployeeName FROM employee\_details")

rows = cur.fetchall()

for row in rows:

if row[0] == username:

emp\_name = row[1]

index = 1

break

if index == 1:

context= {'data':'welcome '+emp\_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("EmployeeAttendance/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 AddEmpAction(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 = 'emp\_attendance',charset='utf8')

with con:

cur = con.cursor()

cur.execute("select employeeID FROM employee\_details")

rows = cur.fetchall()

for row in rows:

if row[0] == empid:

output = ids+" employee already exists"

break

if output == 'none':

db\_connection = pymysql.connect(host='127.0.0.1',port = 3306,user = 'root', password = 'root', database = 'emp\_attendance',charset='utf8')

db\_cursor = db\_connection.cursor()

student\_sql\_query = "INSERT INTO employee\_details(employeeID,empployeeName,phoneNo,designation,emp\_salary) VALUES('"+ids+"','"+name+"','"+phone+"','"+desg+"','"+sal+"')"

db\_cursor.execute(student\_sql\_query)

db\_connection.commit()

url = pyqrcode.create(ids)

url.png('EmployeeAttendance/static/qrcodes/'+ids+'.png', scale = 6)

username = ids

print(db\_cursor.rowcount, "Record Inserted")

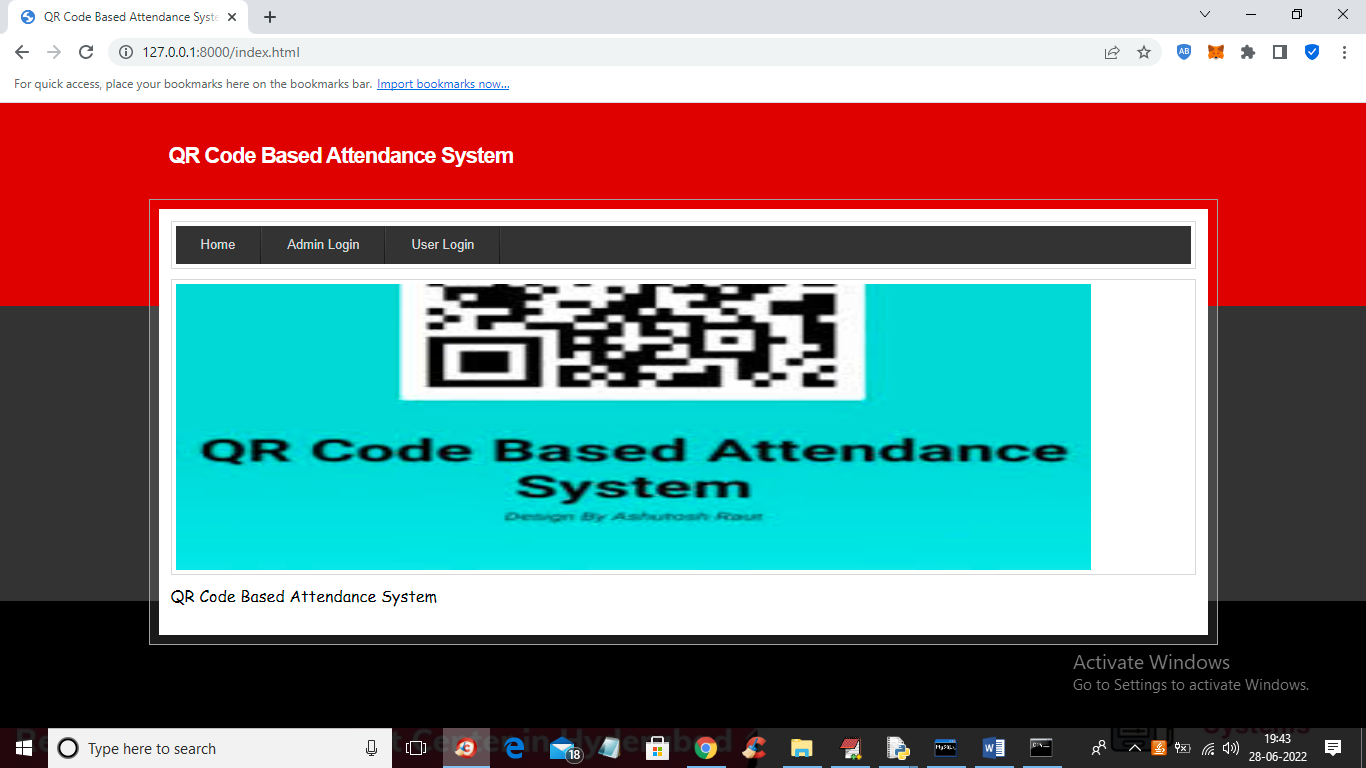
if db\_cursor.rowcount == 1:

output = 'Emp Details Saved with ID : '+ids

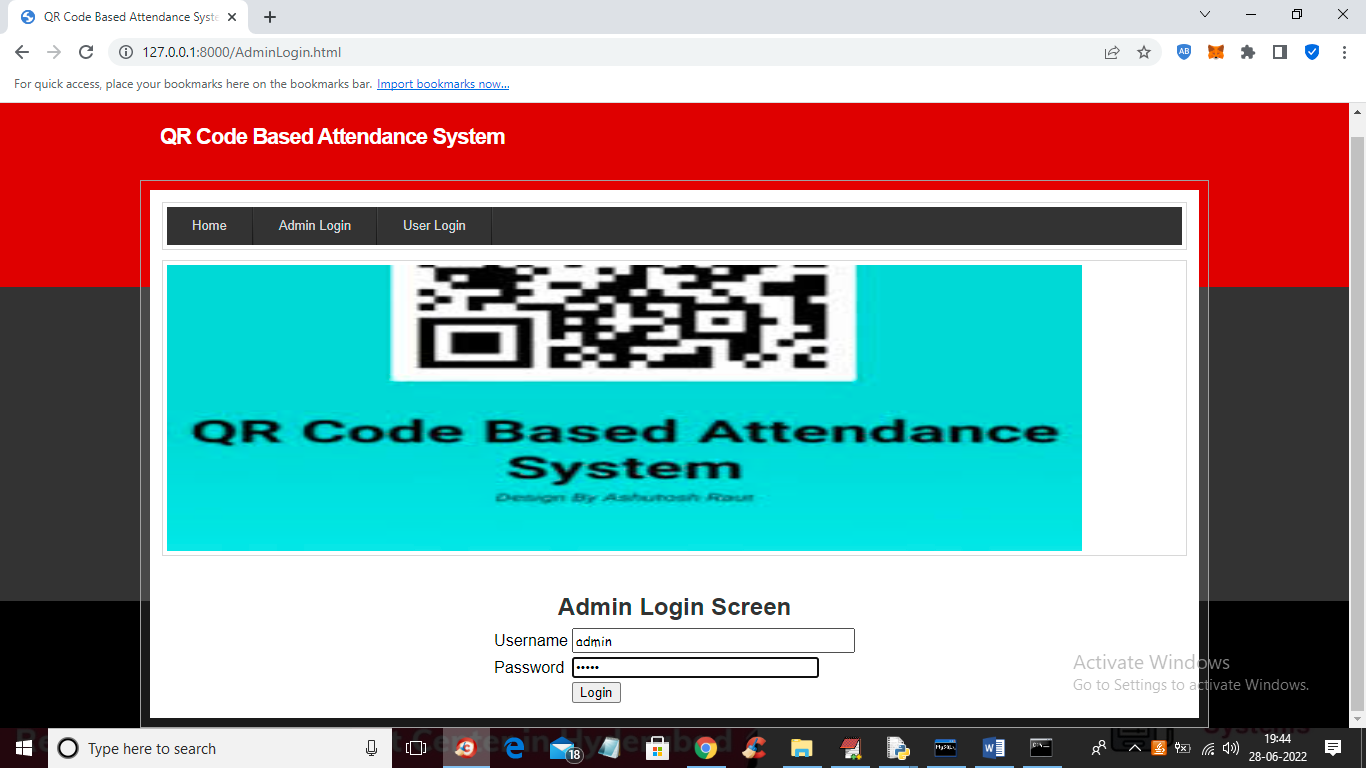
context= {'data':output}

return render(request, 'Download.html', context)

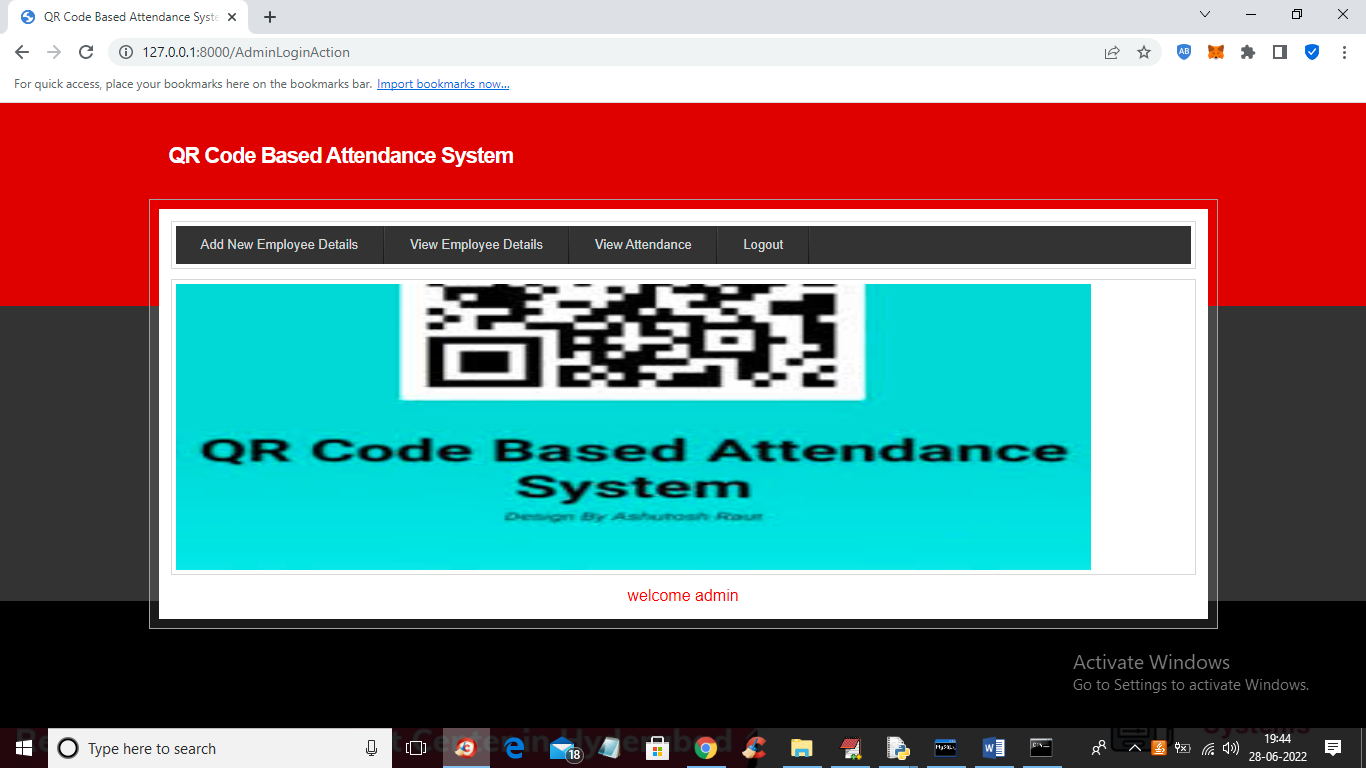
**5.RUNNING RESULTS & SNAPSHOTS :**



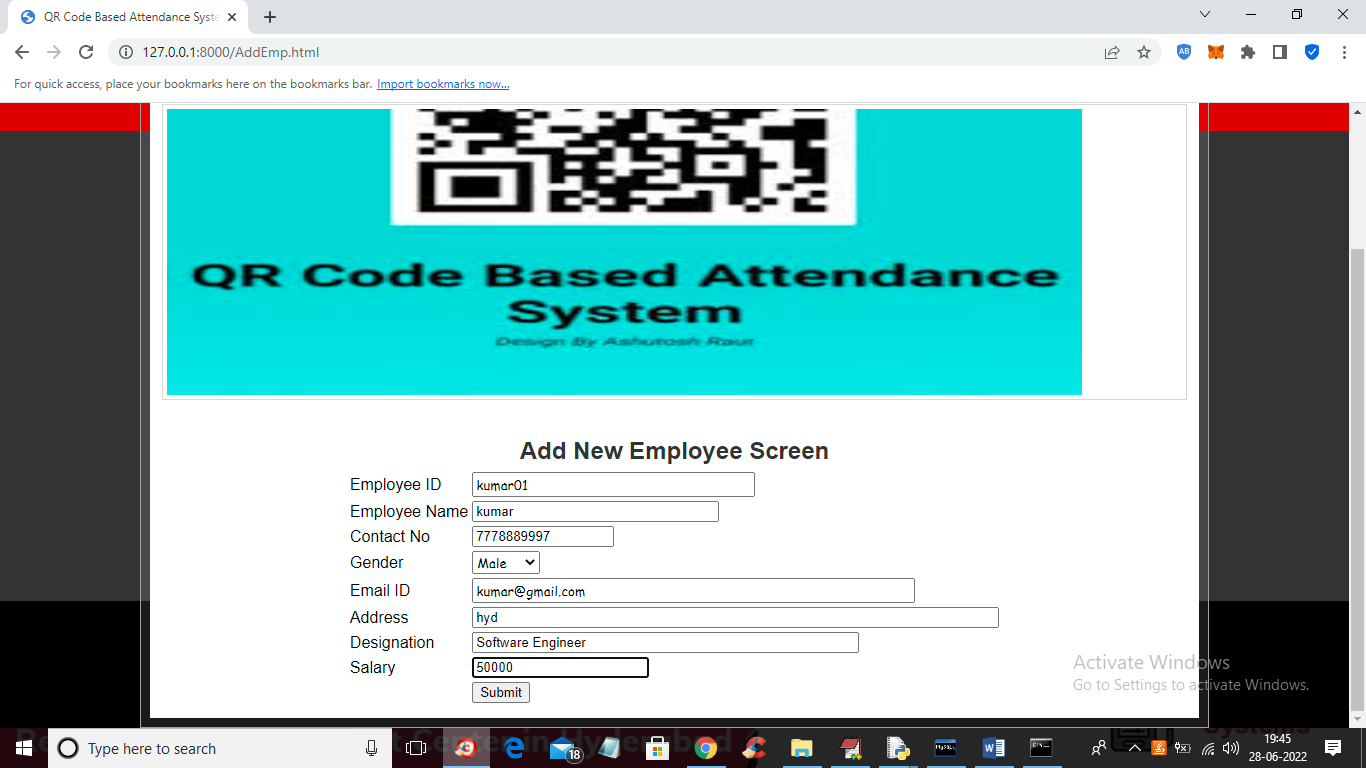
In above screen click on ‘Admin Login’ link to get below login screen



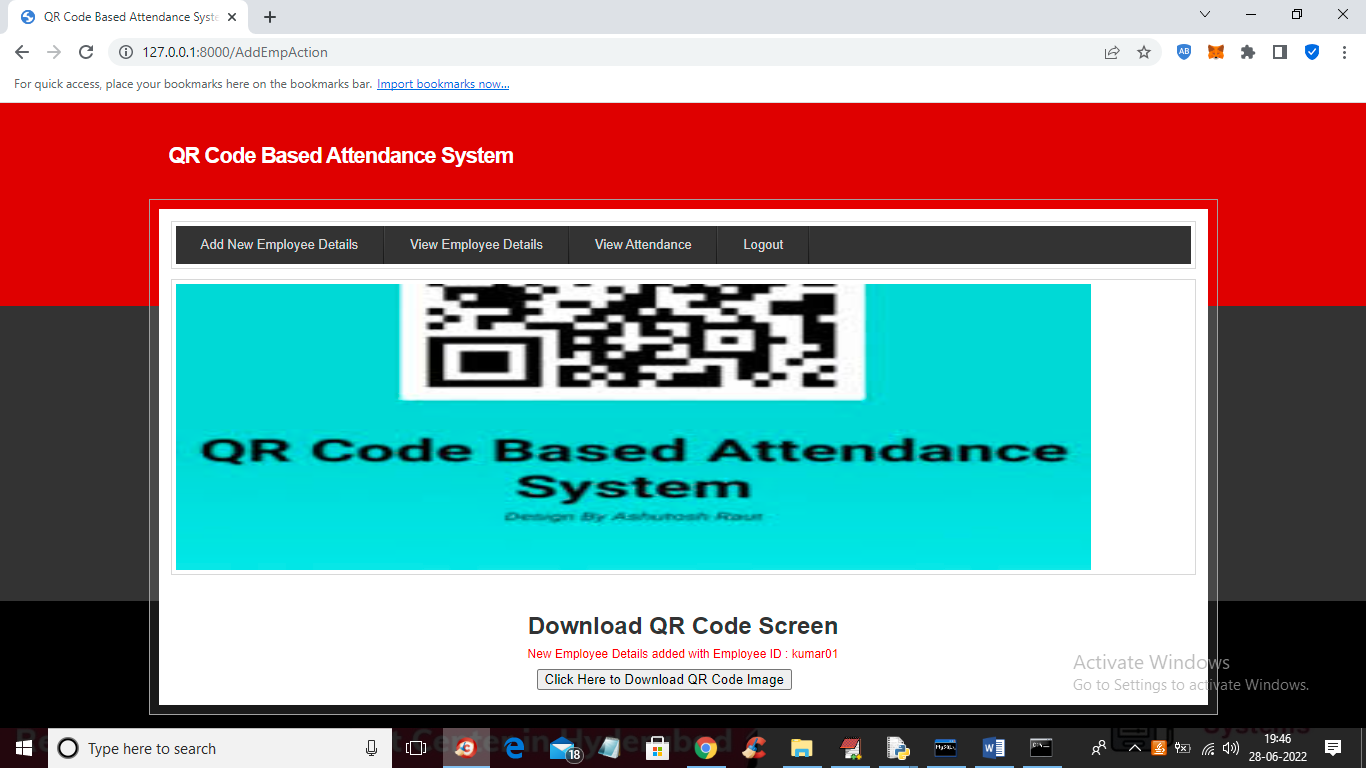
In above screen admin is login and after login will get below screen



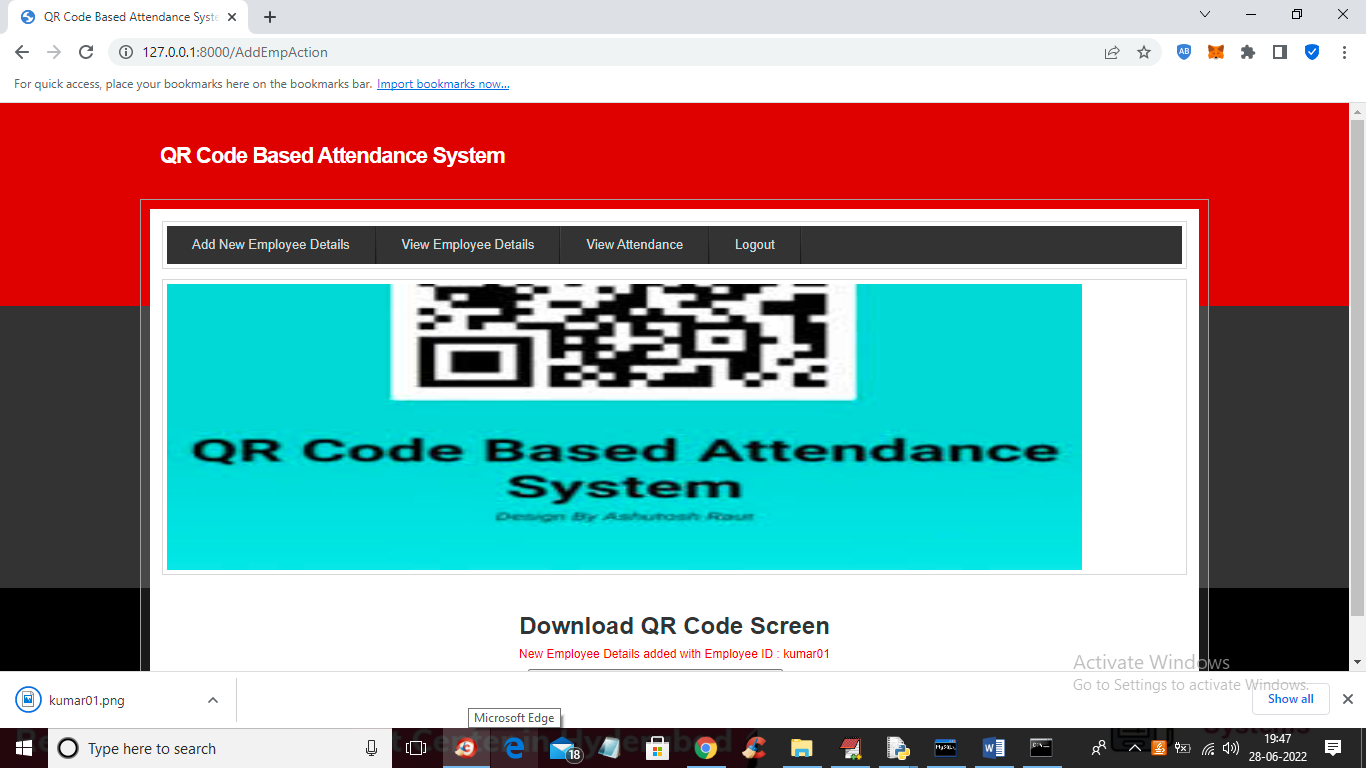
In above screen admin can click on ‘Add New Employee Details’ link to get below screen to add employee details



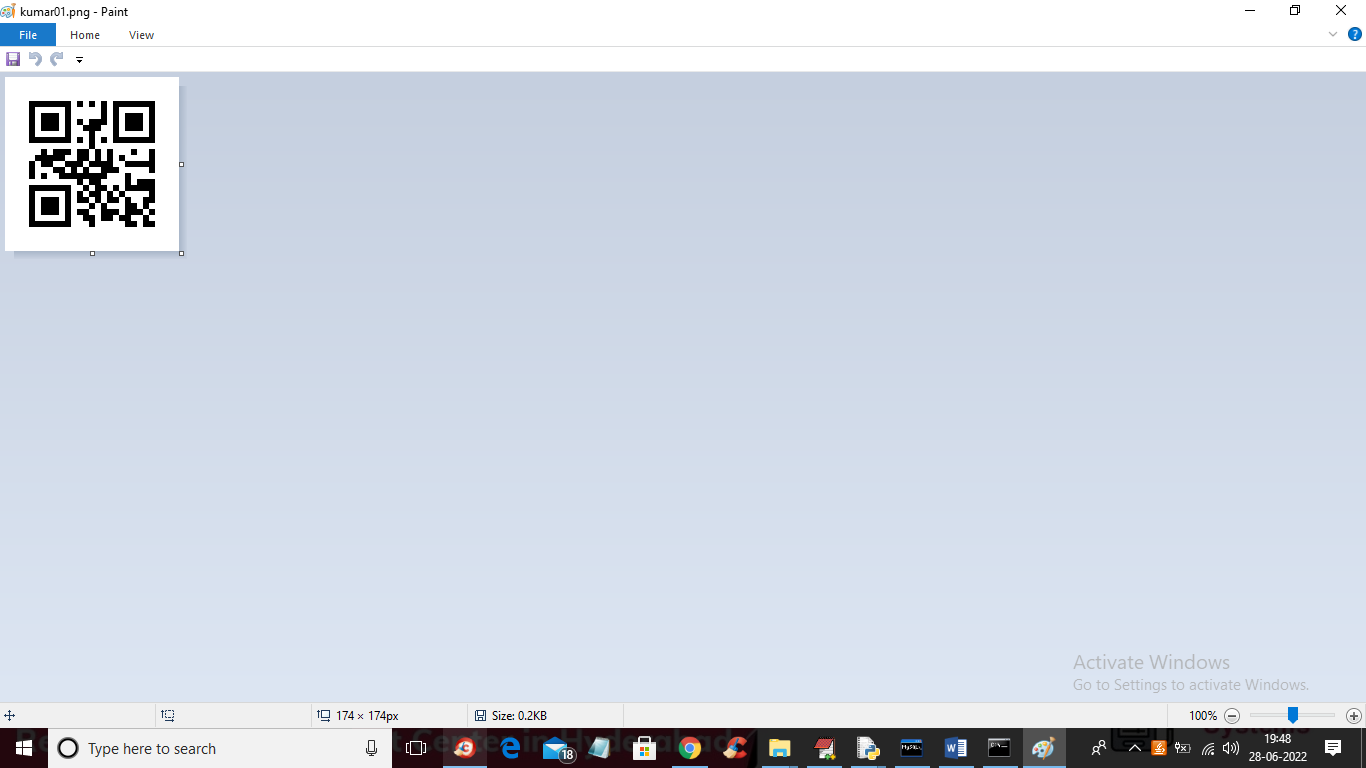
In above screen admin is adding NEW Employee Details and then press button to get below screen



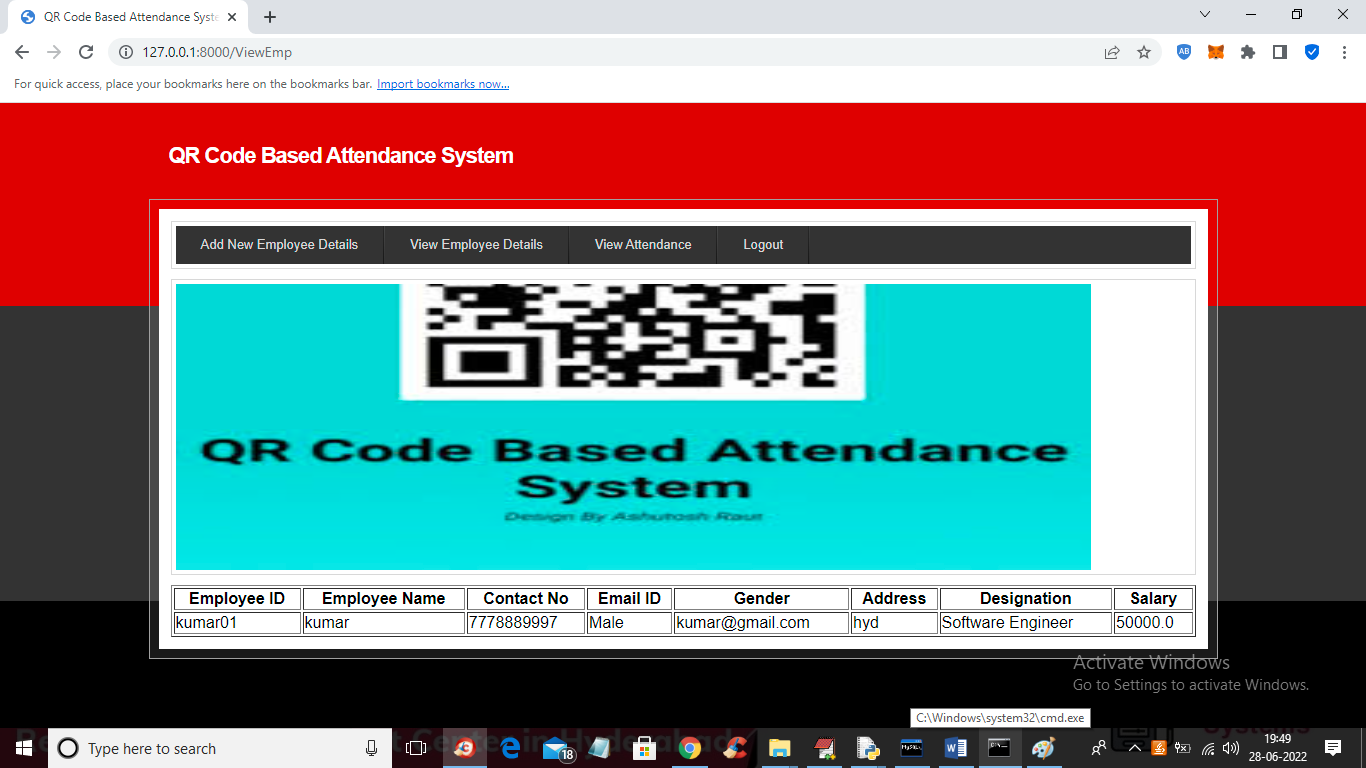
In above screen employee details added and now click on ‘Click Here to Download QR Code Image’ button to download QR image and get below output



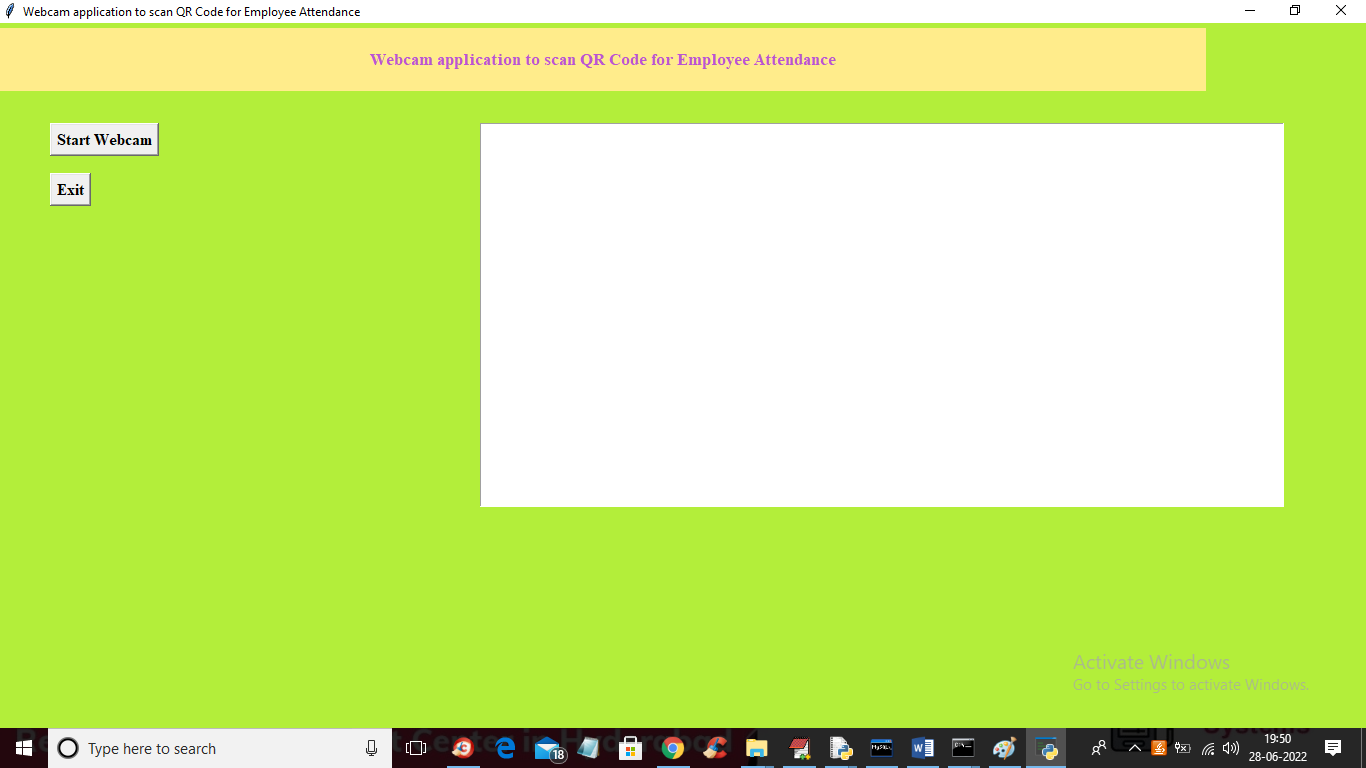
In above screen in browser status bar we can see QR image is downloaded and admin will give this image to employee and he can saved this image in mobile and then can show this image from his mobile to WEBCAM to mark his attendance and now open that image and view QR CODE like below screen



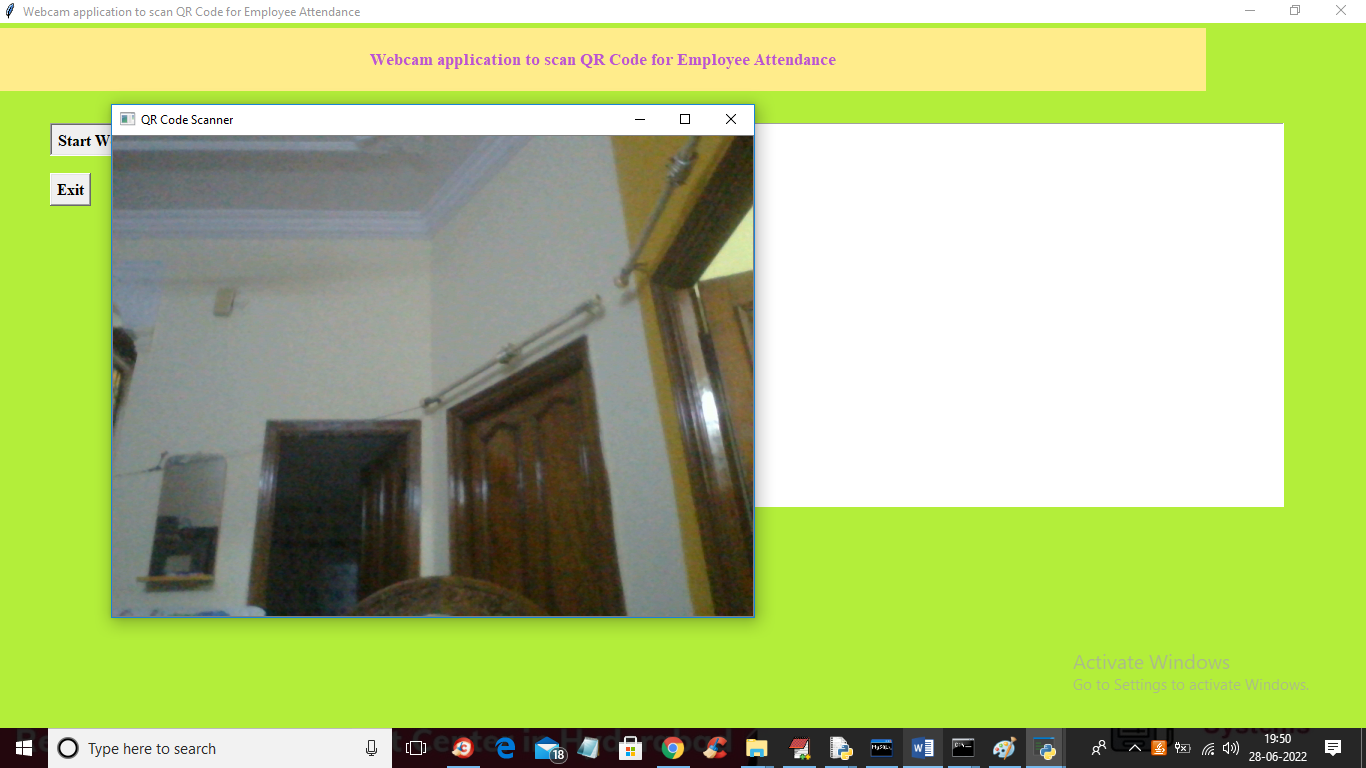
In above screen we can see QR code and now in application click on ‘View Employee Details’ link to get below details



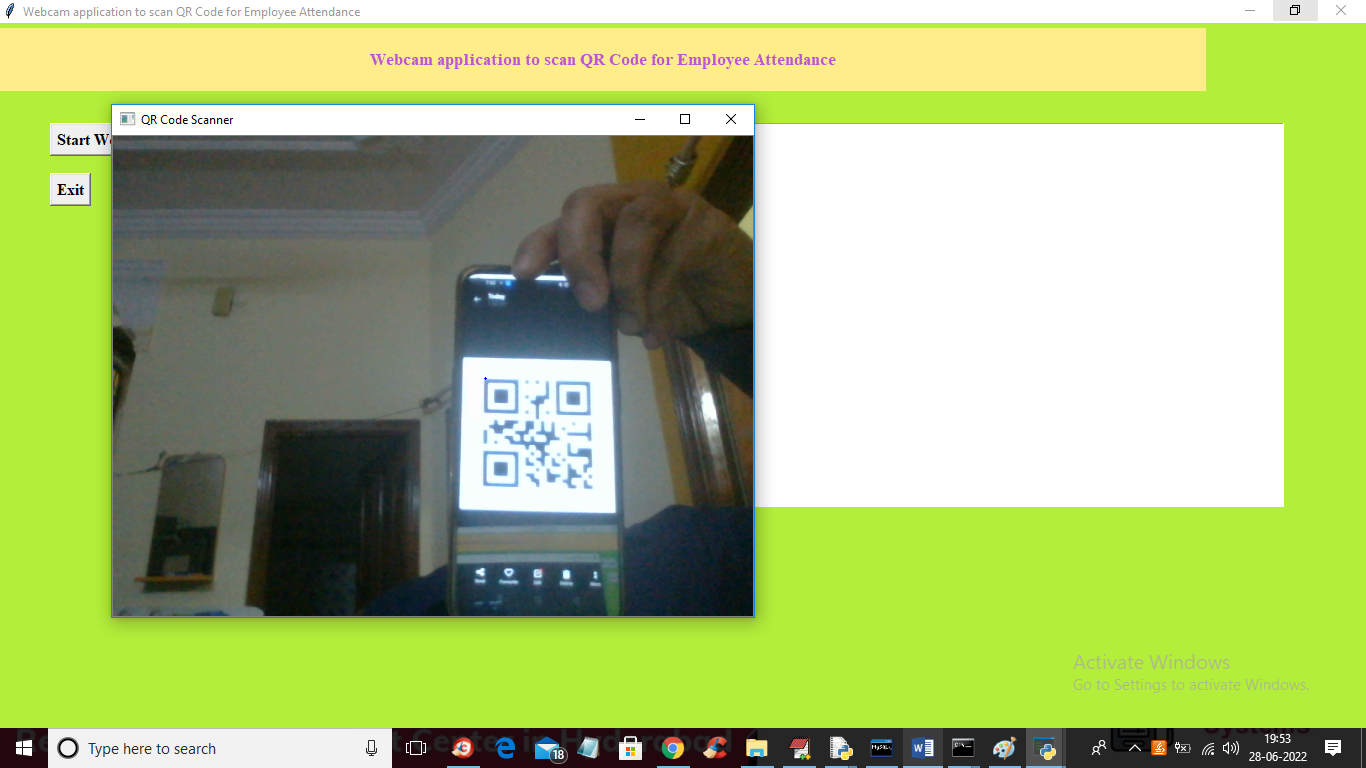
In above screen admin can view all employee details and now to mark attendance double click on ‘RunWebCam.bat’ file to get below screen



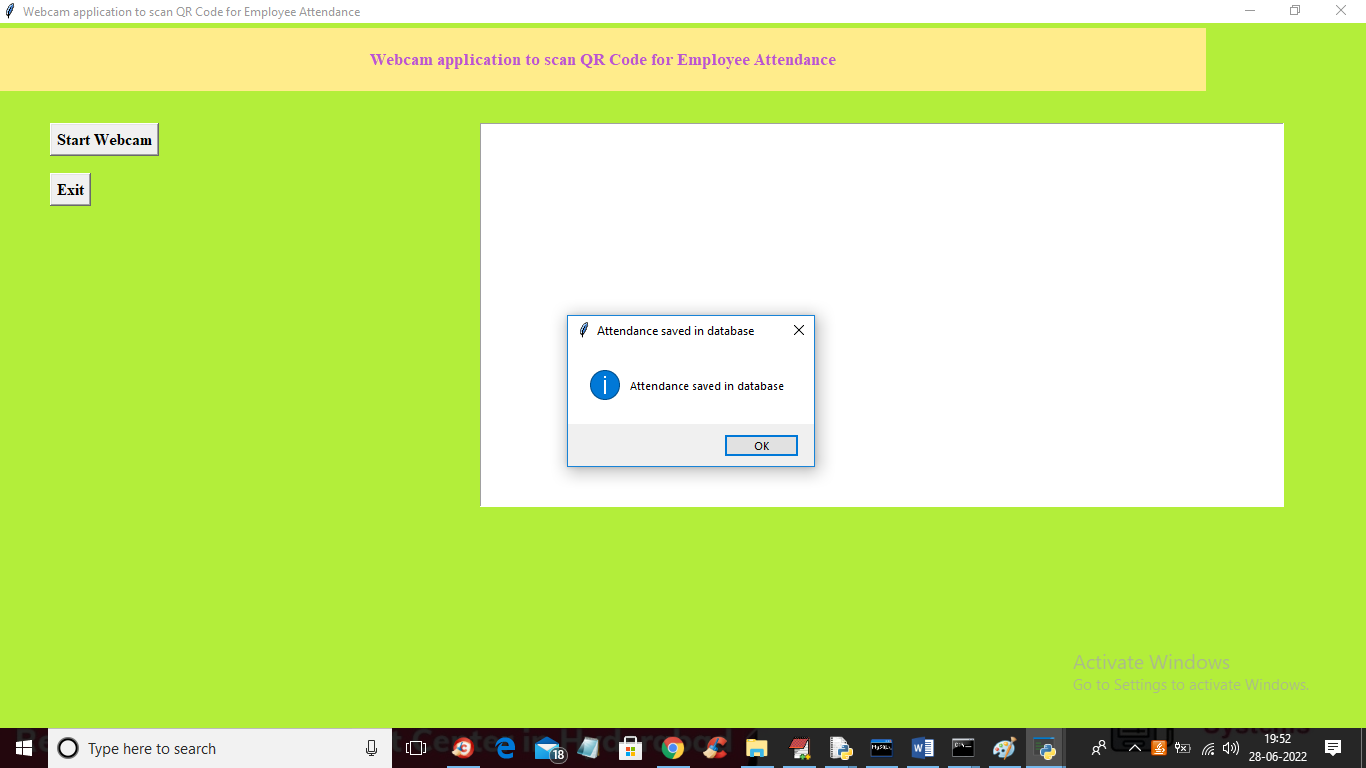
In above screen click on ‘Start Webcam’ button to start web cam and get below screen



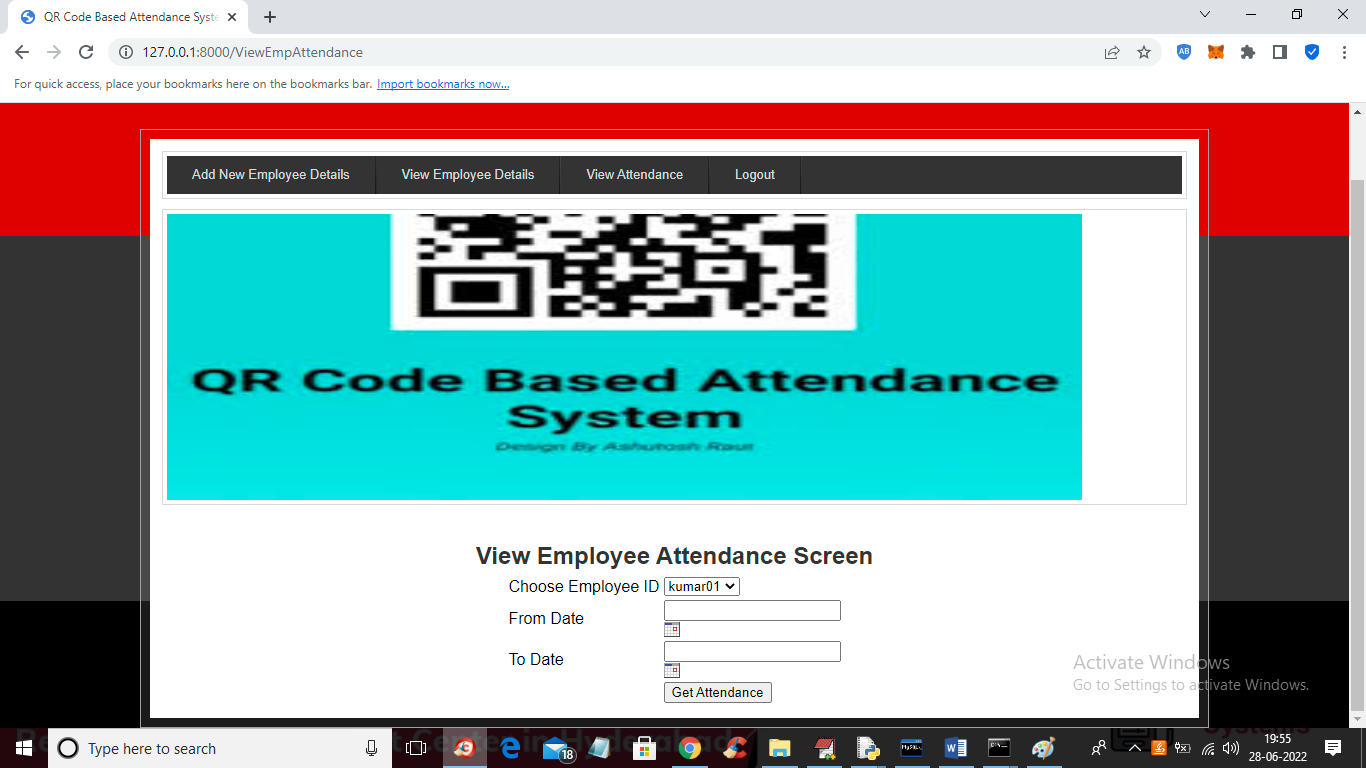
In above screen webcam started and now employee has to show QRCODE from his mobile like below screen and once QR code detected then system will mark attendance



In above screen to webcam I am showing QRCODE and once detected then will get below screen



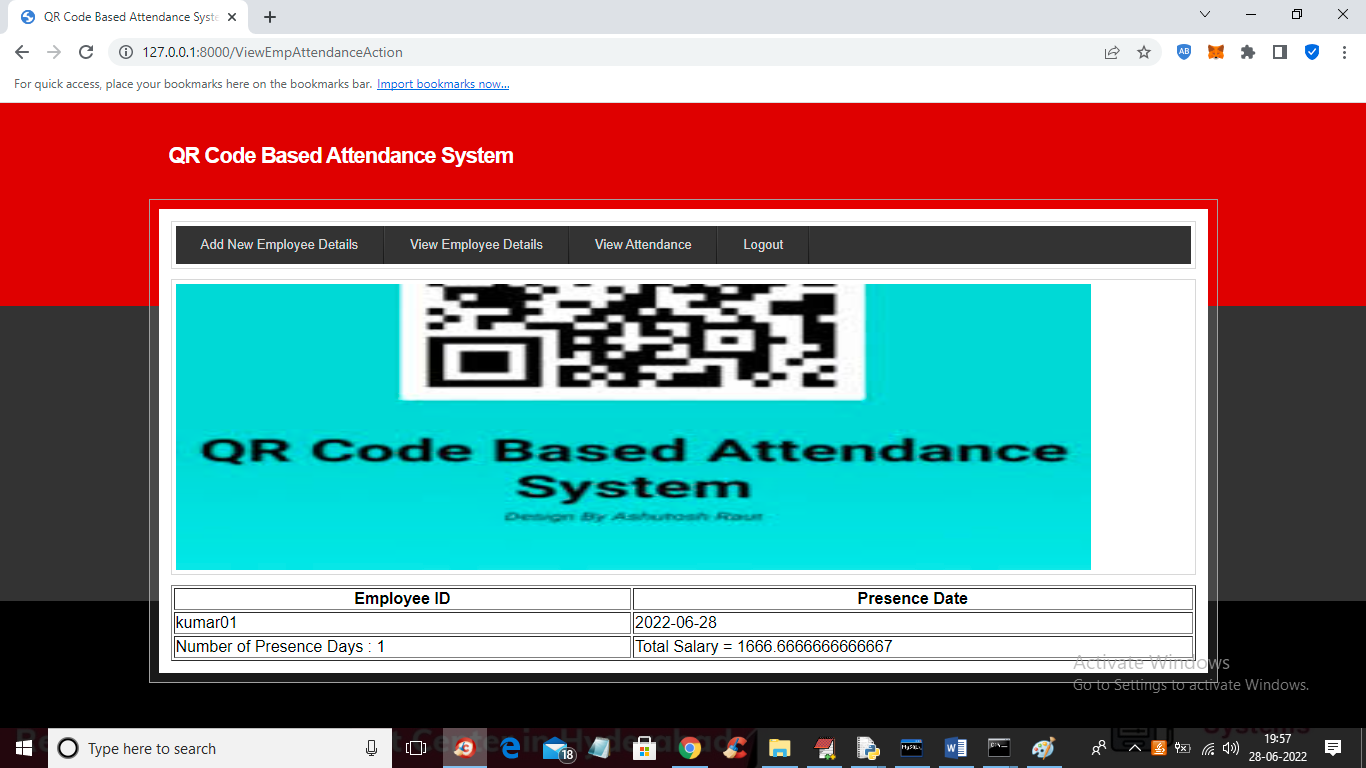
In above screen we got dialog box saying ‘attendance saved in database’ and each employee each day only one time webcam will scan his QRCODE and if he want again then delete all rows from database. Now go to previous application and then click on ‘View Attendance’ link like in below screen



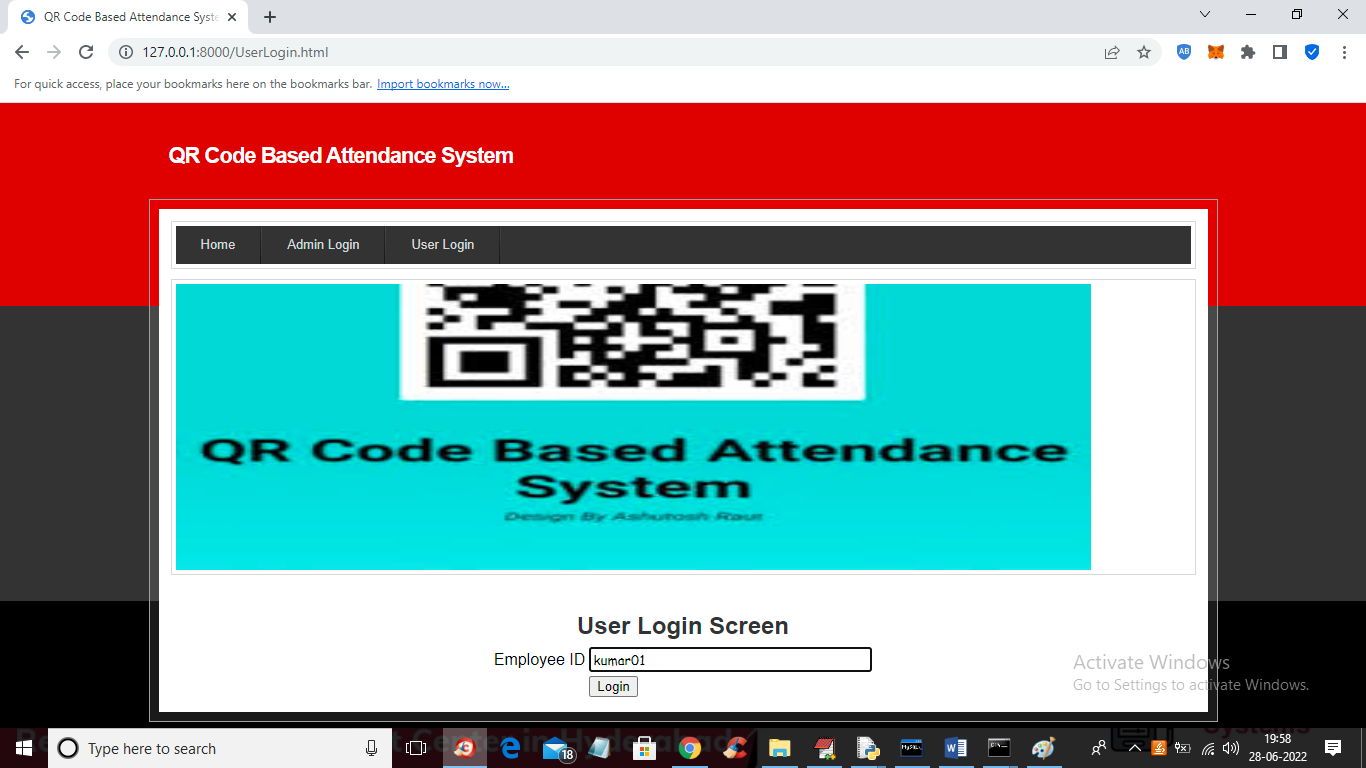
In above view attendance screen admin can view all employee names in drop down box and he can select desired employee name and then choose start and end date like below screen



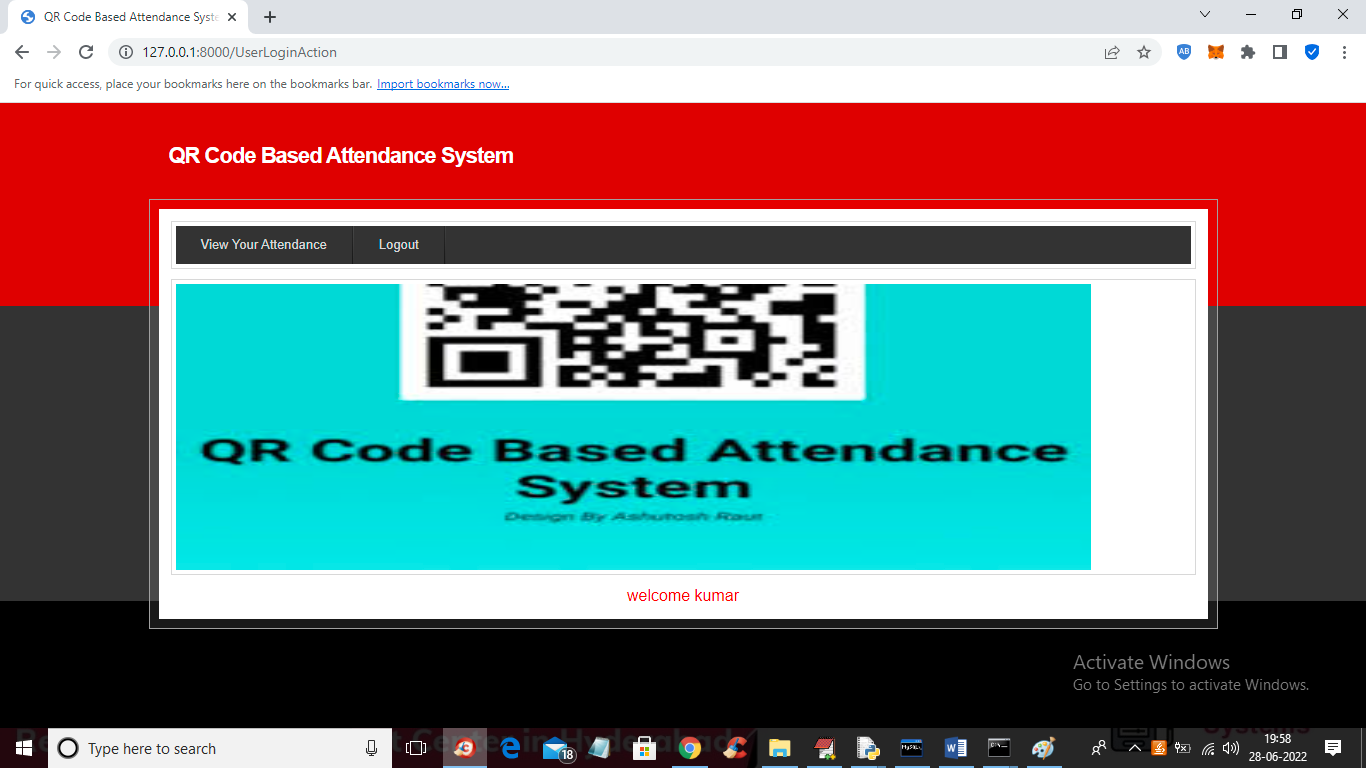
In above screen admin selected employee id and then select start and end date and then press ‘get Attendance’ button to get below screen



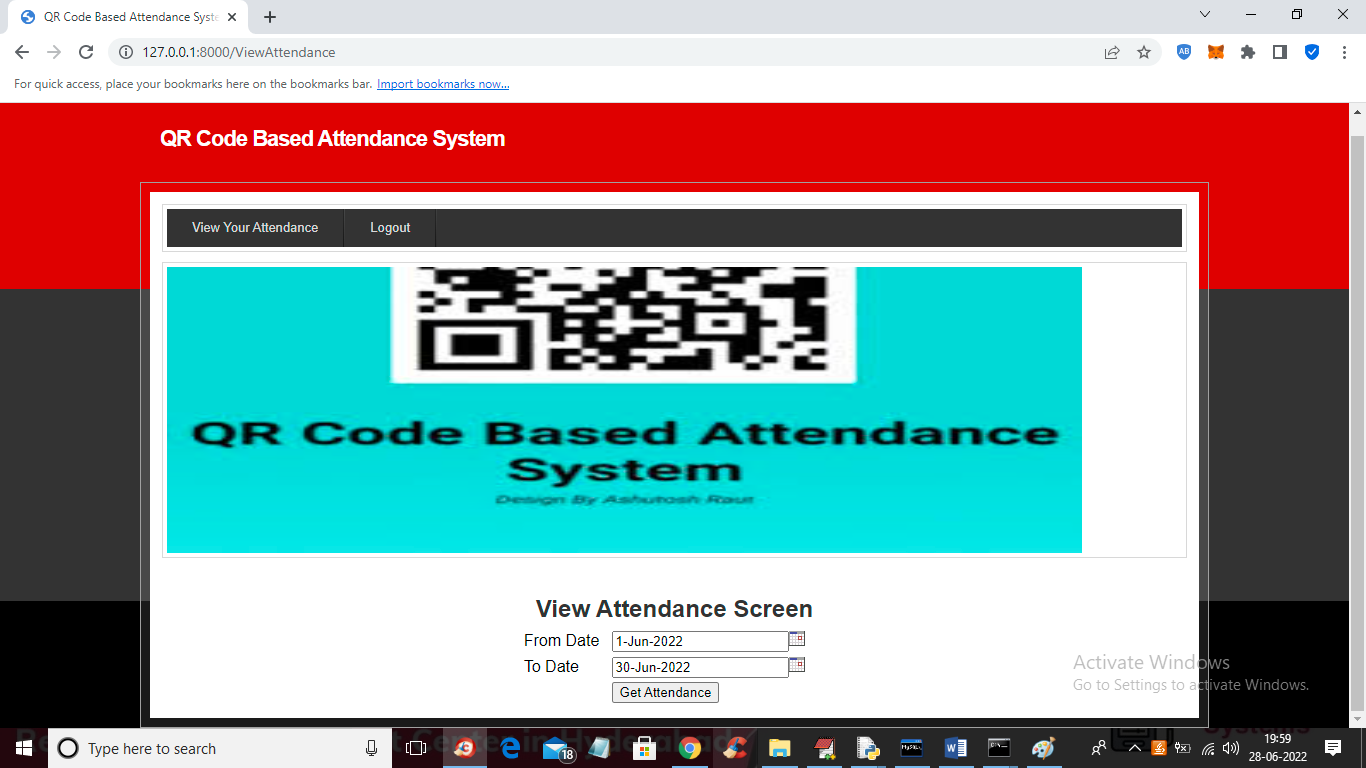
In above screen in first column we can see employee ID and in second column we can see date on which he was present and in last column we can see his payable salary by calculating all present days. Now logout and login as employee



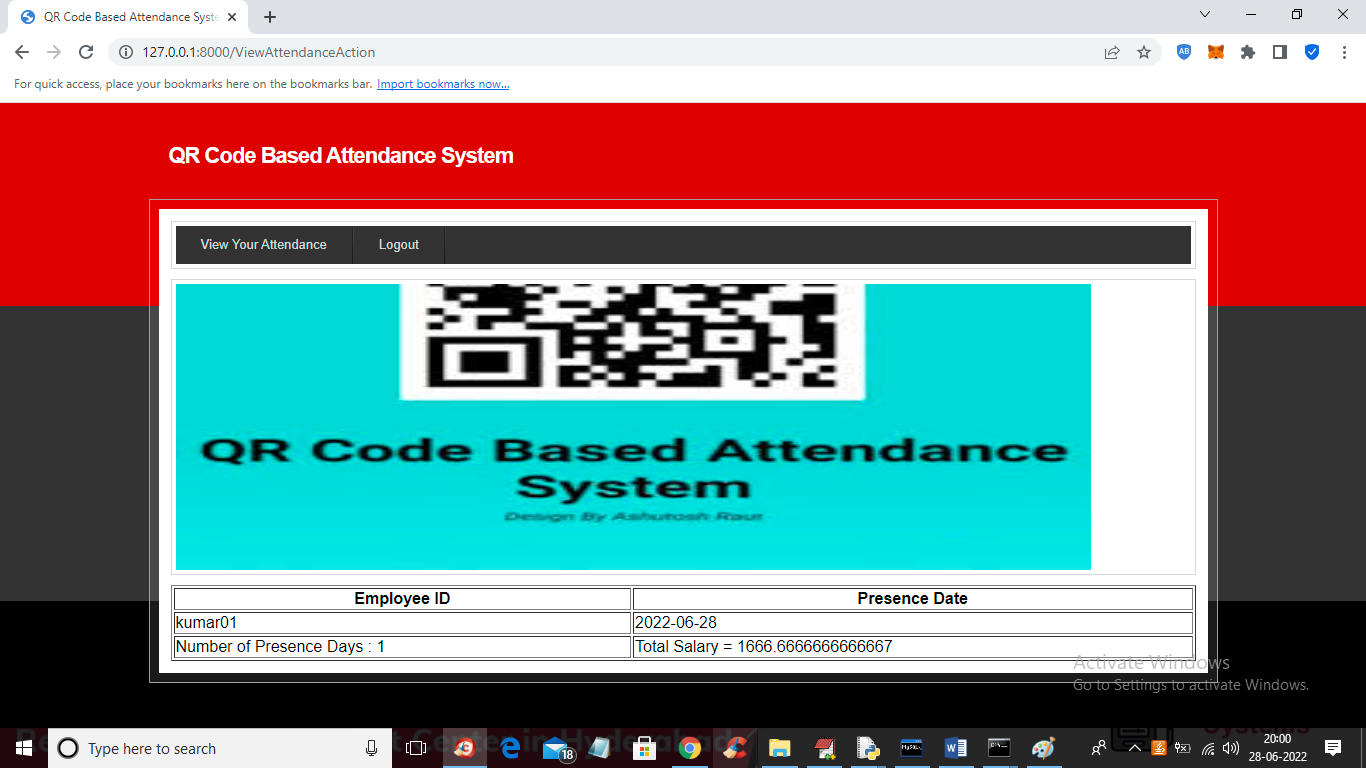
In above screen employee can login by using his ID and then press button to get below screen



In above screen employee can click on ‘View Your Attendance’ link to get below screen



In above screen employee can select start and end date and then press button to view his attendance for selected days



In above screen employee can view all present days date and current payable salary.

Similarly you can add any number of employees and go for attendance and view it