**Smart Attendance Management System**

The project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Sai Charan Teja. J (AP21110011314)**

**Khyathi Devi. K (AP21110011327)**

**Rushita. G (AP21110011328)**



Under the Guidance of

**Shaiju Panchikkil**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**November, 2023**

**Certificate**

Date: 15/11/2023

This is to certify that the work present in this Project entitled **“Smart Attendance Management System”** has been carried out by **Sai Charan Teja. J,** **Khyathi Devi. K, Rushita. G** under our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences.**

**Supervisor**

(Signature)

Prof. Shaiju Panchikkil

Assistant Professor,

Department of Computer Science and Engineering.

**Co-supervisor**

(Signature)

Prof. / Dr.

Designation,

Affiliation.

**Acknowledgments**

We want to express my profound gratitude to Dr Jatindra Kumar Dash, of the CSE department, and Mr. Shaiju Panchikkil for their contributions to the completion of my project titled Smart Attendance Management System.

We would like to express our special thanks to our mentor Mr. Shaiju Panchikkil for the time and efforts he provided throughout the year. Your useful advice and suggestions were helpful to us during the project’s completion. In this aspect, We are eternally grateful to you.

We would like to take this opportunity to express our gratitude to all of our group members. The project would not have been successful without their cooperation and input.

Signature

**Sai Charan Teja. J**

**Khyathi Devi. K**

**Rushita. G**

**Table of Contents**

|  |  |
| --- | --- |
| **Content** | **Pages No** |
| Cover Letter | 1 |
| Certificate | 2 |
| Acknowledgments | 3 |
| Problem Statement | 5 |
| Objectives | 6 |
| Abstract | 7 |
| System Design   1. ER Diagram 2. Schema Diagram Tables | 8-10 |
| Implementation   1. Front end 2. Back end 3. Codes 4. Screenshots | 11 – 46 |
| Conclusion | 47 |
| Future scope | 48 |
| References | 49 |

**Problem Statement**

Manual attendance tracking in educational institutions is inefficient and error-prone. It consumes valuable class time, leads to inaccuracies in record-keeping, and places an unnecessary administrative burden on educators. The need for an automated solution that streamlines attendance management is evident. The Smart Attendance Management System is designed to automate attendance tracking and improve classroom efficiency by using RFID technology. The goal of this project is to create a system that offers precise and efficient attendance recording while reducing manual workload for educators.

**Objectives**

The main objectives of the Smart Attendance Management System are as follows:

* Automated Attendance Tracking: Implement RFID-based automated attendance tracking to eliminate manual processes and enhance the efficiency of recording attendance.
* Efficiency Improvement: Streamline classroom operations by reducing the administrative workload on educators, allowing them to focus on teaching.
* Data-Driven Decision-Making: Enable educators to make informed decisions with comprehensive attendance reports and trends, ultimately improving the educational experience.
* Real-Time Records: Ensure precise, real-time attendance records that are accessible to educators and administrators for immediate insights.

**Abstract**

The educational landscape today faces several challenges, including the inefficient and error-prone process of manual attendance tracking. Traditional methods require instructors to devote valuable classroom time to the laborious task of recording attendance, leading to inaccuracies, resource inefficiencies, and a suboptimal educational experience. These challenges become more pronounced in institutions with larger class sizes, multiple courses, and the need for precise attendance records.

The Smart Attendance Management System is a groundbreaking solution designed to address these issues and revolutionize attendance tracking in educational institutions. This project leverages Radio-Frequency Identification (RFID) technology to automate the attendance process, significantly enhancing efficiency and data accuracy. This report provides an overview of the system's design, implementation, and its potential for future enhancements. It covers the front-end and back-end implementation details, as well as the code for recording attendance. Additionally, relevant HTML and CSS components are discussed.

Our System not only simplifies attendance management but also paves the way for a more efficient, error-free, and data-driven educational environment. It promises to enhance the quality of education by removing the burdens associated with manual attendance tracking and providing educators with valuable insights into student attendance patterns.

**System Design**

**ER Diagram**

student

name

email

student\_id

timetable

Start\_time

End\_time

Course\_name

day

Course\_id

Inst\_id

Inst\_name

Course\_id

Course\_name

Course\_id

Student\_id

status

date

time

Attendance

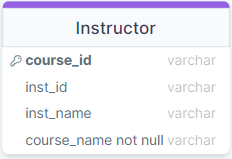
instructor

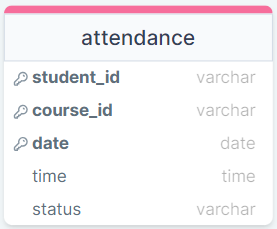
has

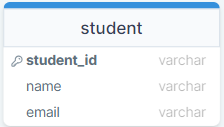
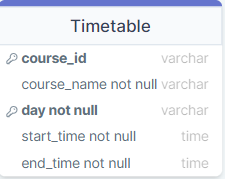
teaches

Enrolls in

**Schema Diagrams**







**Schema Table**

Attendance

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student\_id | Course\_id | date | time | status |

Instructor

|  |  |  |  |
| --- | --- | --- | --- |
| Course\_id | Inst\_id | Inst\_name | Course\_name |

TimeTable

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course\_id | Course\_name | day | Start\_time | End\_time |

Student

|  |  |  |
| --- | --- | --- |
| Student\_id | name | email |

**Implementation:**

**Front end:**

**User Interface Design:**

The front end of the Smart Attendance Management System is meticulously designed to provide a visually appealing and user-friendly interface. HTML is employed to structure the content, CSS for styling, and JavaScript for interactivity. The landing page employs a video background to capture attention, with a dynamically changing navigation bar offering a seamless transition between different sections. The use of Locomotive Scroll enhances the scrolling experience, contributing to the modern and smooth navigation of the application.

**Page Structure:**

The HTML structure is modular, with distinct pages dedicated to specific functionalities. The landing page (index.html) introduces the system, while subsequent pages delve into details such as RFID technology, system options, and quick links. The pages are styled consistently using CSS, ensuring a cohesive and professional appearance throughout the application.

**External Libraries:**

To augment the visual elements, Remixicon and Locomotive Scroll external libraries are integrated. Remixicon provides a set of expressive icons, enhancing the overall design. Locomotive Scroll, a smooth-scrolling library, ensures a fluid and engaging user experience. These libraries contribute to the overall aesthetic appeal and functionality of the front end.

**Back end:**

**Flask Integration:**

The back-end functionality is seamlessly integrated with the front end using Flask, a micro web framework for Python. Flask handles routing and communication between the client-side and server-side components. Through Flask, the Python scripts interact with the web interface, allowing for dynamic updates and real-time data processing.

**Database Interaction:**

The back-end functionality revolves around Python and MySQL to manage the storage and retrieval of attendance data. The mysql.connector library facilitates the interaction with the MySQL database, handling queries and updates seamlessly. The Python script (record\_attendance.py) establishes a connection to the database, ensuring real-time updates of attendance records.

**Attendance Logic:**

The back-end logic involves determining the current course based on the day and time. This is achieved by querying the timetable in the database. The script continuously listens for incoming data from the RFID module, extracts student IDs, and updates attendance records accordingly. This ensures accurate and up-to-date attendance information.

**Error Handling:**

Robust error-handling mechanisms are implemented to address potential issues during database interactions or data processing. This includes catching and logging errors to prevent system failures and maintain the reliability and integrity of the attendance data. The script gracefully handles scenarios such as database connection errors, ensuring uninterrupted functionality.

**Codes:**

**Scanning\_ID\_Card.ino**

#include <ESP8266WiFi.h>

#include <WiFiUdp.h>

#include <MFRC522.h>

#include <SPI.h>

#define RST\_PIN         D3

#define SS\_PIN          D4

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

MFRC522::MIFARE\_Key key;

MFRC522::StatusCode status;

const char\* ssid = "Your ssid";

const char\* password = "Your Password";

const char\* host = " ";  // Change to the IP address of your Python server

unsigned int localPort = 8888;  // Change the port number

WiFiUDP udp;

void setup() {

  Serial.begin(9600);

  SPI.begin();

  udp.begin(localPort);

  mfrc522.PCD\_Init();

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    delay(1000);

    Serial.println("Connecting to WiFi...");

  }

  Serial.println("Connected to WiFi");

  Serial.println(F("Read personal data on a MIFARE PICC:"));

}

void loop() {

  for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;

  byte block;

  byte len;

  if (!mfrc522.PICC\_IsNewCardPresent()) {

    return;

  }

  if (!mfrc522.PICC\_ReadCardSerial()) {

    return;

  }

  Serial.println(F("Card Detected:"));

  byte buffer1[18];

  block = 1;

  len = 18;

  status = mfrc522.PCD\_Authenticate(MFRC522::PICC\_CMD\_MF\_AUTH\_KEY\_A, 1, &key, &(mfrc522.uid));

  if (status != MFRC522::STATUS\_OK) {

    Serial.print(F("Authentication failed: "));

    Serial.println(mfrc522.GetStatusCodeName(status));

    return;

  }

  status = mfrc522.MIFARE\_Read(block, buffer1, &len);

  if (status != MFRC522::STATUS\_OK) {

    Serial.print(F("Reading failed: "));

    Serial.println(mfrc522.GetStatusCodeName(status));

    return;

  }

  char studentID[14];

  for (uint8\_t i = 0; i < 13; i++) {

    studentID[i] = (char)buffer1[i];

  }

  for (uint8\_t i = 0; i < 13; i++) {

    Serial.print(studentID[i]);

  }

  Serial.println();

  studentID[14] = '\0';

  udp.beginPacket(host, localPort);

  udp.write(studentID);

  udp.endPacket();

  delay(1000); // Send a message every second

  mfrc522.PICC\_HaltA();

  mfrc522.PCD\_StopCrypto1();

}

**app.py**

from flask import Flask, render\_template, request, redirect, url\_for, flash

import mysql.connector

from datetime import datetime

import smtplib

from email.mime.multipart import MIMEMultipart  # Add this import

from email.mime.text import MIMEText

import openpyxl

import subprocess

import psutil

import pandas as pd

from openpyxl import Workbook

from openpyxl.utils.dataframe import dataframe\_to\_rows

import os

import matplotlib.pyplot as plt

process\_pid = None

process = None

app = Flask(\_\_name\_\_)

# Database configuration

db\_config = {

    'host': 'localhost',

    'user': 'username of your database',

    'password': 'password of you database',

    'database': 'name of you database',

}

def connect\_to\_database():

    try:

        connection = mysql.connector.connect(\*\*db\_config)

        cursor = connection.cursor()

        print("Connected to the database")

        return connection, cursor

    except mysql.connector.Error as err:

        print(f"Error: {err}")

        return None, None

conn, cursor = connect\_to\_database()

@app.route('/')

def index():

    return render\_template('index.html')

def send\_email\_to\_students(course\_id, minimum\_attendance\_percentage):

    cursor.execute("SELECT s.student\_id,s.name,s.email, (COUNT(CASE WHEN a.status = 'present' THEN 1 ELSE NULL END) \* 100.0 / COUNT(\*)) AS percentage FROM student s join attendance a on s.student\_id=a.student\_id WHERE a.course\_id = %s GROUP BY student\_id HAVING percentage<%s", (course\_id, minimum\_attendance\_percentage))

    results = cursor.fetchall()

    for student\_id, student\_name, email, attendance\_percentage in results:

        smtp\_server = 'smtp.gmail.com'

        smtp\_port = 587

        sender\_email = ‘\*\*\*\*\*\*\*\*\*\*\*\*’

        sender\_password = '\*\*\*\*\*\*\*\*'

        msg = MIMEMultipart()

        msg['From'] = sender\_email

        msg['To'] = email

        msg['Subject'] = 'Attendance Warning'

        body = f"Dear {student\_name},\n\nYour attendance percentage for the course {course\_id}is below the specified percentage.\n\n Please ensure to attend the upcoming classes to improve your attendance percentage."

        msg.attach(MIMEText(body, 'plain'))

        with smtplib.SMTP(smtp\_server, smtp\_port, timeout=40) as server:

            server.starttls()

            server.login(sender\_email, sender\_password)

            server.sendmail(sender\_email, email, msg.as\_string())

        print(f"Email sent to {email}")

@app.route("/send\_mail.html", methods=['GET', 'POST'])

def send\_mail():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        minimum\_attendance\_percentage = float(

            request.form['minimum\_attendance\_percentage'])

        send\_email\_to\_students(course\_id, minimum\_attendance\_percentage)

        status = "Mail sent"

        return render\_template('send\_mail.html', status=status)

    return render\_template('send\_mail.html', status="")

def get\_course\_id(cursor, time):

    day\_of\_week = datetime.now().strftime('%A')

    fetch\_course\_id\_query = "SELECT course\_id FROM timetable WHERE day = %s AND %s BETWEEN start\_time AND end\_time"

    cursor.execute(fetch\_course\_id\_query, (day\_of\_week, time))

    course\_id\_result = cursor.fetchone()

    if course\_id\_result:

        return course\_id\_result[0]

    else:

        return None

def is\_existing\_student(cursor, student\_id, course\_id, date):

    check\_query = "SELECT \* FROM attendance WHERE student\_id = %s AND course\_id = %s AND date = %s"

    cursor.execute(check\_query, (student\_id, course\_id, date))

    existing\_entry = cursor.fetchone()

    if existing\_entry:

        return existing\_entry

    else:

        return False

def mark\_absent(conn, cursor, student\_id, course\_id, date, time):

    if not is\_existing\_student(cursor, student\_id, course\_id, date):

        attendance\_insert\_query = "INSERT INTO attendance (student\_id, course\_id, date, time, status) VALUES (%s, %s, %s, %s, %s)"

        cursor.execute(attendance\_insert\_query,

                       (student\_id, course\_id, date, time, "absent"))

        conn.commit()

def mark\_all\_absent(conn, cursor, course\_id, date):

    current\_time = datetime.now().time().strftime('%H:%M:%S')

    student\_query = "SELECT student\_id FROM student"

    cursor.execute(student\_query)

    students = cursor.fetchall()

    for student in students:

        student\_id = student[0]

        mark\_absent(conn, cursor, student\_id, course\_id, date, current\_time)

def print\_absentees(cursor, course\_id, date):

    print\_absenties\_query = "SELECT student\_id FROM attendance WHERE course\_id = %s AND date = %s AND status = %s"

    cursor.execute(print\_absenties\_query, (course\_id, date, "absent"))

    absent\_entries = cursor.fetchall()

    return absent\_entries

@app.route("/record\_attendance.html", methods=['GET', 'POST'])

def record\_attendance():

    return render\_template('record\_attendance.html')

@app.route('/start\_recording')

def start\_recording():

    global process

    current\_date = datetime.now().date()

    current\_time = datetime.now().time().strftime('%H:%M:%S')

    course\_id = get\_course\_id(cursor, current\_time)

    if course\_id:

        print(course\_id)

        mark\_all\_absent(conn, cursor, course\_id, current\_date)

        print("Recording attendance, press 'q' to terminate")

        process = subprocess.Popen(['python', 'record\_attendance.py'])

        process\_pid = process.pid

        print(course\_id, process\_pid)

    return render\_template('record\_attendance.html')

@ app.route('/stop\_recording')

def stop\_recording():

    current\_date = datetime.now().date()

    current\_time = datetime.now().time().strftime('%H:%M:%S')

    course\_id = get\_course\_id(cursor, current\_time)

    global process

    process.terminate()

    # absentees = ['test']

    # if process\_pid:os.kill(process\_pid, 15)

    absentees = []

    absent\_entries = print\_absentees(cursor, course\_id, current\_date)

    for entry in absent\_entries:

        absentees.append(entry[0])

    print(absentees)

    return render\_template('record\_attendance.html', absentees=absentees)

@app.route('/check\_attendance\_range.html', methods=['GET', 'POST'])

def check\_attendance\_range():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        minimum\_percentage = float(request.form['minimum\_percentage'])

        maximum\_percentage = float(request.form['maximum\_percentage'])

        cursor.execute("SELECT student\_id, (COUNT(CASE WHEN status = 'present' THEN 1 ELSE NULL END) \* 100.0 / COUNT(\*)) AS percentage FROM attendance WHERE course\_id = %s GROUP BY student\_id HAVING percentage BETWEEN %s AND %s",

                       (course\_id, minimum\_percentage, maximum\_percentage))

        results = cursor.fetchall()

        if results:

            students = [{'student\_id': student, 'percentage': round(

                percentage, 2)} for student, percentage in results]

        else:

            students = []

        return render\_template('check\_attendance\_range.html', students=students)

    return render\_template('check\_attendance\_range.html', students=[])

@app.route("/manual\_attendance.html", methods=['GET', 'POST'])

def manual\_attendance():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        date = str(request.form['date\_to\_modify'])

        student\_id = str(request.form['student\_id'])

        get\_status\_query = "SELECT status FROM attendance WHERE student\_id = %s AND course\_id = %s AND date = %s;"

        cursor.execute(get\_status\_query, (student\_id, course\_id, date))

        status\_result = cursor.fetchall()

        return render\_template('manual\_attendance.html', status=status\_result)

    return render\_template('manual\_attendance.html', status=[])

@app.route("/modify\_attendance", methods=['GET', 'POST'])

def modify\_attendance():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        date = str(request.form['date\_to\_modify'])

        student\_id = str(request.form['student\_id'])

        new\_status = request.form['status']

        print("hii", date)

        update\_attendance\_query = "UPDATE attendance SET status = %s WHERE student\_id = %s AND course\_id = %s AND date = %s;"

        cursor.execute(update\_attendance\_query,

                       (new\_status, student\_id, course\_id, date))

        print("Attendance modified successfully!")

        get\_status\_query = "SELECT status FROM attendance WHERE student\_id = %s AND course\_id = %s AND date = %s;"

        cursor.execute(get\_status\_query, (student\_id, course\_id, date))

        status\_result = cursor.fetchall()

        print(student\_id, "\t", status\_result[0][0])

        return render\_template('manual\_attendance.html')

    return render\_template('manual\_attendance.html')

@app.route("/particular\_day\_attendance.html", methods=['GET', 'POST'])

def check\_particular\_day\_attendance():

    if request.method == 'POST':

        date = str(request.form['date\_to\_check'])

        course\_id = request.form['course\_id']

        cursor.execute(

            "select student\_id, status from attendance where date = %s AND course\_id = %s", (date, course\_id))

        result = cursor.fetchall()

        present\_students = []

        absent\_students = []

        for student\_id, status in result:

            if status == "present":

                present\_students.append(student\_id)

            elif status == "absent":

                absent\_students.append(student\_id)

        students = []

        students.append(present\_students)

        students.append(absent\_students)

        return render\_template('particular\_day\_attendance.html', students=students)

    return render\_template('particular\_day\_attendance.html', students=[])

@app.route("/particular\_student\_attendance.html", methods=['GET', 'POST'])

def check\_particular\_student\_attendance():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        date = str(request.form['date\_to\_check'])

        student\_id = request.form['student\_id']

        existing\_record = is\_existing\_student(

            cursor, student\_id, course\_id, date)

        if existing\_record:

            students = []

            print(existing\_record[4])

            students.append("")

            students.append(existing\_record[4])

            return render\_template('particular\_student\_attendance.html', students=students)

        else:

            students = []

            print("No record found")

            students.append("student id not found")

            students.append("")

            return render\_template('particular\_student\_attendance.html', students=students)

    return render\_template('particular\_student\_attendance.html')

@app.route("/attendance\_trend\_chart.html", methods=['GET', 'POST'])

def attendance\_trend\_chart():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        start\_date = request.form['start\_date']

        end\_date = request.form['end\_date']

        date\_range\_attendance\_query = """

                    SELECT date,

                    SUM(CASE WHEN status = 'Present' THEN 1 ELSE 0 END) AS present\_count,

                    SUM(CASE WHEN status = 'Absent' THEN 1 ELSE 0 END) AS absent\_count

                FROM

                    attendance

                WHERE

                    course\_id = %s AND

                    date BETWEEN %s AND %s

                GROUP BY

                    date

                ORDER BY

                    date;

                """

        cursor.execute(date\_range\_attendance\_query,

                       (course\_id, start\_date, end\_date))

        data = cursor.fetchall()

        df = pd.DataFrame(

            data, columns=['date', 'present\_count', 'absent\_count'])

        img\_path = "static/Images/plot.png"

        plt.figure(figsize=(8, 6))

        plt.plot(df['date'], df['present\_count'], label='Present', marker='o')

        plt.plot(df['date'], df['absent\_count'], label='Absent', marker='o')

        plt.xlabel('Date')

        plt.ylabel('Number of Students')

        plt.title(f'Attendance from {start\_date} to {end\_date}')

        plt.subplots\_adjust(bottom=0.2)

        plt.xticks(rotation=45)

        plt.legend()

        plt.grid(True)

        for i, row in df.iterrows():

            plt.text(row['date'], row['present\_count']+2,

                     str(row['present\_count']), ha='center', va='top', fontsize=8)

            plt.text(row['date'], row['absent\_count']+2,

                     str(row['absent\_count']), ha='center', va='top', fontsize=8)

        plt.savefig(img\_path, format='png')

        plt.close()

        return render\_template('attendance\_trend\_chart.html', image\_path="Images/plot.png")

    return render\_template('attendance\_trend\_chart.html', image\_path="")

@app.route("/create\_attendance\_excel.html", methods=['GET', 'POST'])

def create\_attendance\_excel():

    if request.method == 'POST':

        start\_date = str(request.form['start\_date'])

        end\_date = str(request.form['end\_date'])

        course\_id = request.form['course\_id']

        start\_date = start\_date.replace(":", "-")

        end\_date = end\_date.replace(":", "-")

        file\_name = f"{course\_id}attendance{start\_date}to{end\_date}.xlsx"

        query = """

        SELECT

            a.student\_id,

            s.name AS student\_name,

            s.email AS student\_email,

            COUNT(\*) AS total\_classes\_conducted,

            SUM(CASE WHEN a.status = 'present' THEN 1 ELSE 0 END) AS classes\_attended

        FROM attendance a

        JOIN student s ON a.student\_id = s.student\_id

        WHERE a.course\_id = %s AND a.date BETWEEN %s AND %s

        GROUP BY a.student\_id;

        """

        cursor.execute(query, (course\_id, start\_date, end\_date))

        attendance\_data = cursor.fetchall()

        # print("hiii",attendance\_data)

        if not attendance\_data:

            status = "No attendance records found for the specified date range."

            return render\_template('create\_attendance\_excel.html', status=status)

        else:

            df = pd.DataFrame(

                attendance\_data,

                columns=[

                    "Student ID",

                    "Student Name",

                    "Student Email",

                    "Total Classes Conducted",

                    "Classes Attended"

                ]

            )

            df["Attendance Percentage"] = (

                df["Classes Attended"] / df["Total Classes Conducted"]) \* 100

            df = df.sort\_values(by="Attendance Percentage", ascending=False)

            wb = Workbook()

            ws = wb.active

            for col\_num, column\_title in enumerate(df.columns, 1):

                ws.cell(row=1, column=col\_num, value=column\_title)

            for record in df.to\_records(index=False):

                ws.append(list(record))

            wb.save(file\_name)

            status = (

                f"Student attendance performance excel sheet for {course\_id} between {start\_date} and {end\_date} has been saved to '{file\_name}'.")

            return render\_template('create\_attendance\_excel.html', status=status)

    return render\_template('create\_attendance\_excel.html', status=[])

@app.route("/generate\_monthly\_attendance.html", methods=['GET', 'POST'])

def generate\_monthly\_attendance():

    if request.method == 'POST':

        course\_id = request.form['course\_id']

        year = request.form['year']

        month = request.form['month']

        start\_date = f"{year}-{month}-01"

        end\_date = f"{year}-{month}-31"

        report\_query = """

        SELECT student\_id, COUNT(\*) AS total\_classes,

            SUM(CASE WHEN status = 'present' THEN 1 ELSE 0 END) AS attended\_classes

        FROM attendance

        WHERE course\_id = %s AND date BETWEEN %s AND %s

        GROUP BY student\_id;

        """

        cursor.execute(report\_query, (course\_id, start\_date, end\_date))

        report\_data = cursor.fetchall()

        report = []

        for result in report\_data:

            row = []

            row.append(result[0])

            row.append(result[1])

            row.append(result[2])

            attendance\_percentage = (result[2] / result[1]) \* 100

            attendance\_percentage = "{:.2f}".format(attendance\_percentage)

            row.append(attendance\_percentage)

            report.append(row)

        # print(report)

        return render\_template('generate\_monthly\_attendance.html', report=report)

    return render\_template('generate\_monthly\_attendance.html')

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

**record\_attendance.py**

import socket

import mysql.connector

from datetime import datetime

db\_config = {

    'host': 'localhost',

    'user': 'root',

    'password': 'Nani@01012004',

    'database': 'dbms\_project',

}

UDP\_IP = "0.0.0.0"

UDP\_PORT = 8888

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sock.bind((UDP\_IP, UDP\_PORT))

try:

    conn = mysql.connector.connect(\*\*db\_config)

    cursor = conn.cursor()

    print("Connected to the database")

except mysql.connector.Error as err:

    print(f"Error: {err}")

    exit()

def get\_course\_id():

    date = datetime.now()

    day\_of\_week = date.strftime('%A')

    time = date.strftime("%H:%M:%S")

    fetch\_course\_id\_query = "SELECT course\_id FROM timetable WHERE day = %s AND %s BETWEEN start\_time AND end\_time"

    cursor.execute(fetch\_course\_id\_query, (day\_of\_week, time))

    course\_id\_result = cursor.fetchone()

    if course\_id\_result:

        return course\_id\_result[0]

    else:

        return None

course\_id = get\_course\_id()

while True:

    data, addr = sock.recvfrom(1024)

    student\_id = data.decode()

    student\_id = student\_id[:-1]

    print(student\_id)

    now = datetime.now()

    date = now.strftime("%Y-%m-%d")

    time = now.strftime("%H:%M:%S")

    status = "present"

    try:

        cursor.execute("UPDATE attendance set status=%s,time=%s where student\_id=%s and course\_id=%s and date=%s",

                       (status, time, student\_id, course\_id, date))

        conn.commit()

        print("Recorded attendance for Student ID:", student\_id)

    except mysql.connector.Error as err:

        print(f"Error: {err}")

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="../static/style.css">

    <link href="https://cdn.jsdelivr.net/npm/remixicon@3.2.0/fonts/remixicon.css" rel="stylesheet">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/locomotive-scroll@3.5.4/dist/locomotive-scroll.css">

</head>

<body>

    <div id="main">

        <div data-scroll data-scroll-speed="-10" id="page1">

            <nav>

                <img src="../static/Images/Srm logo.png" alt="">

                <div id="right-nav">

                    <button>Options</button>

                    <button>About us</button>

                    <button><i class="ri-menu-fill"></i></button>

                </div>

            </nav>

            <div class="bottom-page1">

                <h1> Smart Attendance<br> Management System</h1> `

                <div class="bottom-page1-inner">

                    <h4>Efficiency in Every Tap,<br> Precision in Every Record.</h4>

                </div>

            </div>

            <video src="../static/Videos/vid.mp4" autoplay loop muted></video>

        </div>

        <div id="page2">

            <h2>Attendance Management System using RFID</h2>

            <h1>Smart Attendance Management System employs RFID technology for automated tracking, optimizing classroom

                efficiency, and ensuring accurate record-keeping.

            </h1>

        </div>

        <div id="page3">

            <h1>Choose an Option</h1>

            <div class="options-container">

                <a href="record\_attendance.html" class="option">Scan ID <br> Cards</a>

                <a href="manual\_attendance.html" class="option">Manual <br>Attendance</a>

                <a href="check\_attendance\_range.html" class="option">Details of Students <br>With in

                    Particular

                    <br>Attendance

                    range</a>

                <a href="generate\_monthly\_attendance.html" class="option">Monthly Report</a>

                <a href="particular\_student\_attendance.html" class="option">Check the <br> attendance of a <br>

                    student</a>

                <a href="particular\_day\_attendance.html" class="option">Attendance Report <br>of a particular

                    <br>day</a>

                <a href="send\_mail.html" class="option">Send alert mails</a>

                <a href="attendance\_trend\_chart.html" class="option">Attendance trend <br> Chart</a>

                <a href="create\_attendance\_excel.html" class="option">Get Attendance <br> detials <br> in Excel

                    Fromat</a>

            </div>

        </div>

        <div id="page4">

            <h3>About the RFID Technology</h3>

            <h1>RFID technology plays a pivotal role in this system by providing secure, contactless student

                identification, effectively automating attendance management, reducing manual workload, and ensuring

                real-time, precise record-keeping. This innovation streamlines the entire process, making it more

                efficient, accurate, and user-friendly, ultimately improving the educational experience for both

                educators and students.</h1>

        </div>

        <div id="page5">

            <div class="left5">

                <h1>What is <br> RFID?</h1>

            </div>

            <div class="right5">

                <div class="right5-center"></div>

            </div>

        </div>

        <div id="page6">

            <div class="right6">

                <div class="right6-inner">

                    <h1>RFID Technology: Simplifying Access Control</h1>

                    <p>Radio-Frequency Identification (RFID) is at the heart of our innovative access control system.

                        Magma's RFID technology enables seamless management of physical access to buildings and spaces.

                    </p>

                </div>

                <div class="right6-inner">

                    <h1>How RFID Works</h1>

                    <p>RFID works by using radio waves to communicate between a reader and an RFID tag. The tag contains

                        a unique identification code, which can be linked to individuals, objects, or assets. When an

                        RFID tag comes within range of the reader, it sends its information wirelessly, allowing for

                        quick and convenient access control.</p>

                </div>

                <div class="right6-inner">

                    <h1>Enhancing Security and Efficiency</h1>

                    <p>With RFID technology, you can enhance security by controlling who has access to your building or

                        specific areas within it. It offers an efficient and contactless way to manage access rights,

                        making it a valuable tool for modern security and asset management.</p>

                </div>

            </div>

        </div>

        <div id="page7">

            <h1>Quick links</h1>

        </div>

        <div id="page8">

            <div class="page8-inner">

                <h1>Home</h1>

                <i class="ri-arrow-right-up-line"></i>

                <div class="center8"></div>

            </div>

            <div class="page8-inner">

                <h1>Git Hub</h1>

                <i class="ri-arrow-right-up-line"></i>

                <div class="center8"></div>

            </div>

            <div class="page8-inner">

                <h1>SRM AP</h1>

                <i class="ri-arrow-right-up-line"></i>

                <div class="center8"></div>

            </div>

        </div>

    </div>

    <script src="https://cdn.jsdelivr.net/npm/locomotive-scroll@3.5.4/dist/locomotive-scroll.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.1/gsap.min.js"

        integrity="sha512-qF6akR/fsZAB4Co1QDDnUXWnaQseLGXoniuSuSlPQK6+aWhlMZcHzkasCSlnWoe+TJuudlka1/IQ01Dnhgq95g=="

        crossorigin="anonymous" referrerpolicy="no-referrer"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.1/ScrollTrigger.min.js"

        integrity="sha512-IHDCHrefnBT3vOCsvdkMvJF/MCPz/nBauQLzJkupa4Gn4tYg5a6VGyzIrjo6QAUy3We5HFOZUlkUpP0dkgE60A=="

        crossorigin="anonymous" referrerpolicy="no-referrer"></script>

    <script src="../static/script.js"></script>

</body>

</html>

**style.css**

\* {

    margin: 0%;

    padding: 0%;

    box-sizing: border-box;

}

html,

body {

    height: 100%;

    width: 100%;

}

#main {

    position: relative;

    overflow: hidden;

    /\* background-color: #0a3cce; \*/

    background-color: black;

}

@font-face {

    font-family: a;

    src: url(Fonts/jost-variable.ttf);

}

@font-face {

    font-family: b;

    src: url(Fonts/KFOlCnqEu92Fr1MmEU9fBBc4\ \(1\).ttf);

}

@font-face {

    font-family: c;

    src: url(Fonts/KFOmCnqEu92Fr1Mu4mxK\ \(1\).ttf);

}

#page1 {

    height: 100vh;

    width: 100vw;

    position: relative;

}

#page1>video {

    height: 100%;

    width: 100%;

    object-fit: cover;

}

#page1>nav {

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 0px 30px;

    position: absolute;

    height: 12vh;

    width: 100vw;

}

#page1>nav>img {

    margin-top: 0.8vw;

    width: 5%;

}

#right-nav>button {

    padding: 10px 20px;

    border-radius: 50px;

    border: 1px solid #fff;

    background-color: transparent;

    color: #fff;

    font-family: a;

    font-size: 15px;

}

.bottom-page1 {

    position: absolute;

    bottom: 17%;

    height: 35vh;

    width: 60vw;

    left: 10%;

}

.bottom-page1>h1 {

    font-family: a;

    font-size: 5vw;

    font-weight: 100;

    line-height: 1;

    color: #fff;

}

.bottom-page1-inner {

    position: absolute;

    bottom: 0%;

    height: 35%;

    width: 100%;

}

.bottom-page1-inner {

    display: flex;

    align-items: center;

    justify-content: space-between;

    font-family: a;

}

.bottom-page1-inner>h4 {

    font-size: 1.8vw;

    font-weight: 100;

    color: #fff;

}

#page2 {

    display: flex;

    align-items: start;

    font-family: a;

    justify-content: center;

    flex-direction: column;

    height: 100vh;

    width: 100vw;

    position: relative;

    padding: 0vw 8vw;

    color: #dadada69;

    /\* background-color: #0a3cce; \*/

    background-color: black;

}

#page2>h2 {

    margin-bottom: 3vw;

    font-weight: 100;

    color: #fff;

}

#page2>h1 {

    font-weight: 100;

    line-height: 1.3;

    width: 90%;

    font-size: 4vw;

    color: #dadada69;

}

#page3 {

    display: flex;

    align-items: center;

    justify-content: center;

    flex-direction: column;

    position: relative;

    height: 100vh;

    width: 100vw;

    background-color: black;

    font-family: a;

}

#page3>h1 {

    font-size: 3rem;

    font-weight: 100;

    color: #fff;

}

.options-container {

    display: flex;

    flex-wrap: wrap;

    justify-content: center;

    gap: 20px;

    margin-top: 20px;

}

.option {

    width: calc(18% - 20px);

    /\* Square-shaped, and gap included \*/

    padding: 20px;

    background-color: black;

    color: #fff;

    font-family: a;

    text-align: center;

    font-size: 1.5rem;

    border: 1px solid white;

    border-radius: 10px;

    cursor: pointer;

    transition: background-color 0.3s, transform 0.2s;

}

.option {

    height: 0;

    padding-bottom: 20%;

    text-decoration: none;

}

.option:hover {

    background-color: white;

    color: black;

    transform: scale(1.05);

}

#page4 {

    display: flex;

    align-items: start;

    justify-content: center;

    position: relative;

    height: 100vh;

    width: 100vw;

    /\* background-color: #0a3cce; \*/

    background-color: black;

    flex-direction: column;

    font-family: a;

}

#page4>h3 {

    margin-left: 15vw;

    font-weight: 100;

    color: #fff;

    margin-bottom: 2vw;

}

#page4>h1 {

    margin-left: 15vw;

    font-size: 3vw;

    width: 70%;

    font-weight: 100;

    color: #ffffff53;

}

#page5 {

    display: flex;

    position: relative;

    height: 100vh;

    width: 100vw;

    /\* background-color: #0a3cce; \*/

    background-color: black;

}

.left5 {

    height: 100%;

    width: 40%;

    position: relative;

    font-family: a;

}

.left5>h1 {

    position: absolute;

    top: 40%;

    right: 30%;

    transform: translateY(-50%);

    font-size: 5vw;

    font-weight: 100;

    color: #fff;

    line-height: 1;

}

.right5 {

    height: 100%;

    width: 60%;

    position: relative;

}

.right5-center {

    height: 50%;

    width: 85%;

    border-radius: 10px;

    position: absolute;

    top: 50%;

    transform: translateY(-50%);

    background-image: url(Images/RFID\_img.jpg);

    background-size: cover;

    left: 0%;

}

#page6 {

    position: relative;

    height: 100vh;

    width: 100vw;

    /\* background-color: #0a3cce; \*/

    background-color: black;

}

.right6 {

    height: 100%;

    width: 60%;

    position: relative;

    left: 40%;

}

.right6-inner {

    display: flex;

    align-items: start;

    flex-direction: column;

    height: 33.3%;

    width: 100%;

    font-family: a;

    color: #fff;

}

.right6-inner>h1 {

    font-size: 2vw;

}

.right6-inner>p {

    margin-top: 2vw;

    font-size: 1.3vw;

    width: 80%;

}

#page7 {

    position: relative;

    height: 40vh;

    width: 100vw;

    background-color: white;

    color: black;

    font-family: a;

    padding: 7vw 10vw;

}

#page7>h1 {

    top: 5%;

    font-size: 5vw;

    line-height: 1;

    font-weight: 100;

}

#page8 {

    position: relative;

    height: 60vh;

    width: 100vw;

    background-color: #000;

}

.page8-inner {

    position: relative;

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding: 0vw 5vw;

    font-family: a;

    height: 33.3%;

    width: 100%;

    color: white;

    border-top: .5px solid #ffffff5c;

    border-bottom: .5px solid #ffffff48;

}

.page8-inner>i {

    font-weight: 100;

    font-size: 2.4vw;

    position: relative;

    z-index: 9999;

}

.page8-inner>h1 {

    font-size: 3vw;

    font-weight: 100;

    position: relative;

    z-index: 9999;

}

.center8 {

    height: 0%;

    width: 100%;

    background-color: rgb(111, 111, 111);

    /\* color: black; \*/

    position: absolute;

    left: 50%;

    top: 50%;

    transform: translate(-50%, -50%);

    transition: all ease .5s;

}

.page8-inner:hover .center8 {

    height: 100%;

    color: black;

}

**script.js**

function loco(){

  gsap.registerPlugin(ScrollTrigger);

// Using Locomotive Scroll from Locomotive https://github.com/locomotivemtl/locomotive-scroll

const locoScroll = new LocomotiveScroll({

el: document.querySelector("#main"),

smooth: true

});

// each time Locomotive Scroll updates, tell ScrollTrigger to update too (sync positioning)

locoScroll.on("scroll", ScrollTrigger.update);

// tell ScrollTrigger to use these proxy methods for the "#main" element since Locomotive Scroll is hijacking things

ScrollTrigger.scrollerProxy("#main", {

scrollTop(value) {

  return arguments.length ? locoScroll.scrollTo(value, 0, 0) : locoScroll.scroll.instance.scroll.y;

}, // we don't have to define a scrollLeft because we're only scrolling vertically.

getBoundingClientRect() {

  return {top: 0, left: 0, width: window.innerWidth, height: window.innerHeight};

},

// LocomotiveScroll handles things completely differently on mobile devices - it doesn't even transform the container at all! So to get the correct behavior and avoid jitters, we should pin things with position: fixed on mobile. We sense it by checking to see if there's a transform applied to the container (the LocomotiveScroll-controlled element).

pinType: document.querySelector("#main").style.transform ? "transform" : "fixed"

});

// each time the window updates, we should refresh ScrollTrigger and then update LocomotiveScroll.

ScrollTrigger.addEventListener("refresh", () => locoScroll.update());

// after everything is set up, refresh() ScrollTrigger and update LocomotiveScroll because padding may have been added for pinning, etc.

ScrollTrigger.refresh();

}

loco()

var clutter = "";

document.querySelector("#page2>h1").textContent.split("").forEach(function(dets){

  clutter += `<span>${dets}</span>`

  document.querySelector("#page2>h1").innerHTML = clutter;

})

gsap.to("#page2>h1>span",{

  scrollTrigger:{

      trigger:`#page2>h1>span`,

      start:`top bottom`,

      end:`bottom top`,

      scroller:`#main`,

      scrub:.5,

  },

  stagger:.2,

  color:`#fff`

})

var clutter = "";

document.querySelector("#page4>h1").textContent.split("").forEach(function(dets){

  clutter += `<span>${dets}</span>`

  document.querySelector("#page4>h1").innerHTML = clutter;

})

gsap.to("#page4>h1>span",{

scrollTrigger:{

    trigger:`#page4>h1>span`,

    start:`top bottom`,

    end:`bottom top`,

    scroller:`#main`,

    scrub:.5,

},

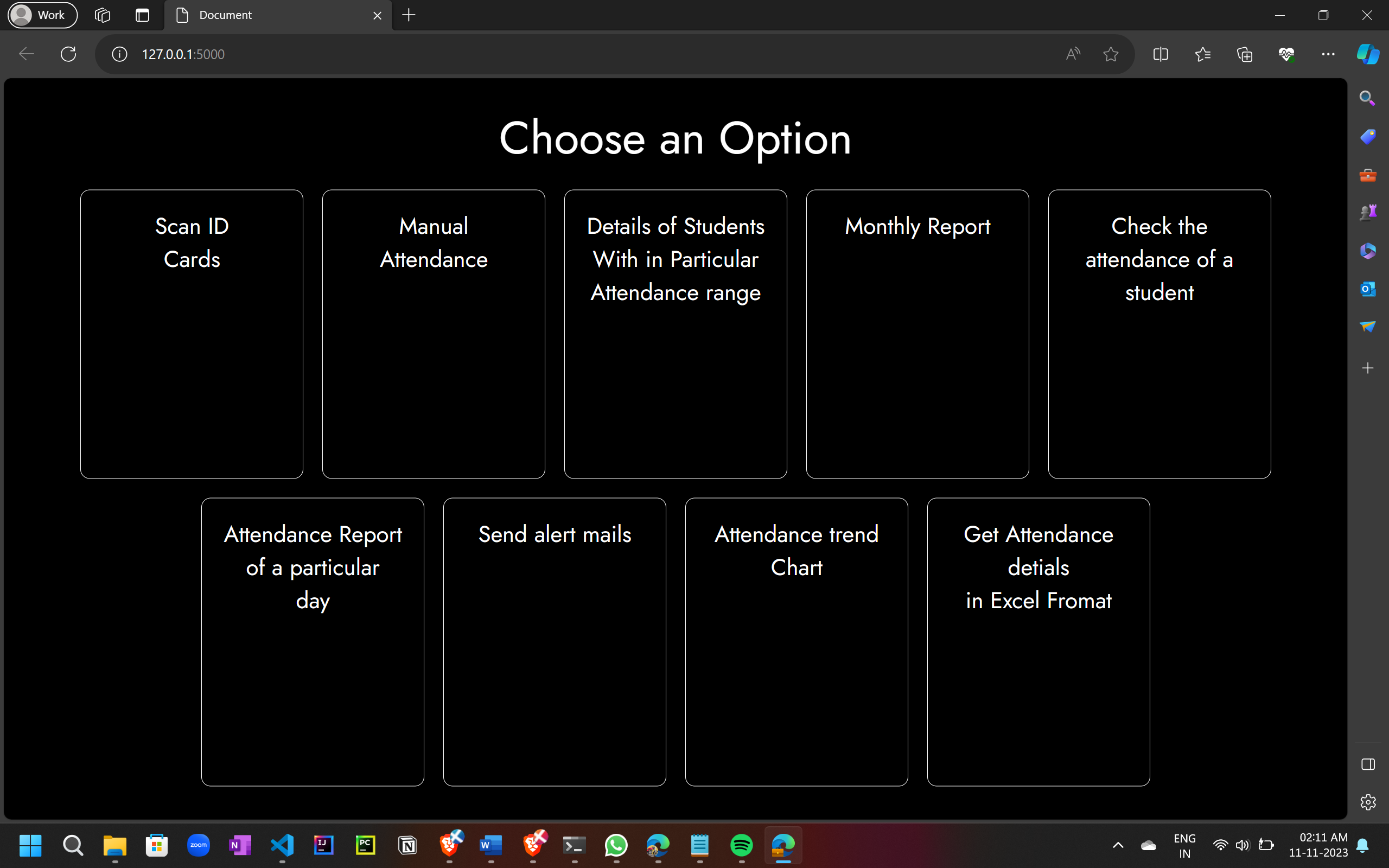
stagger:.2,

color:`#fff`

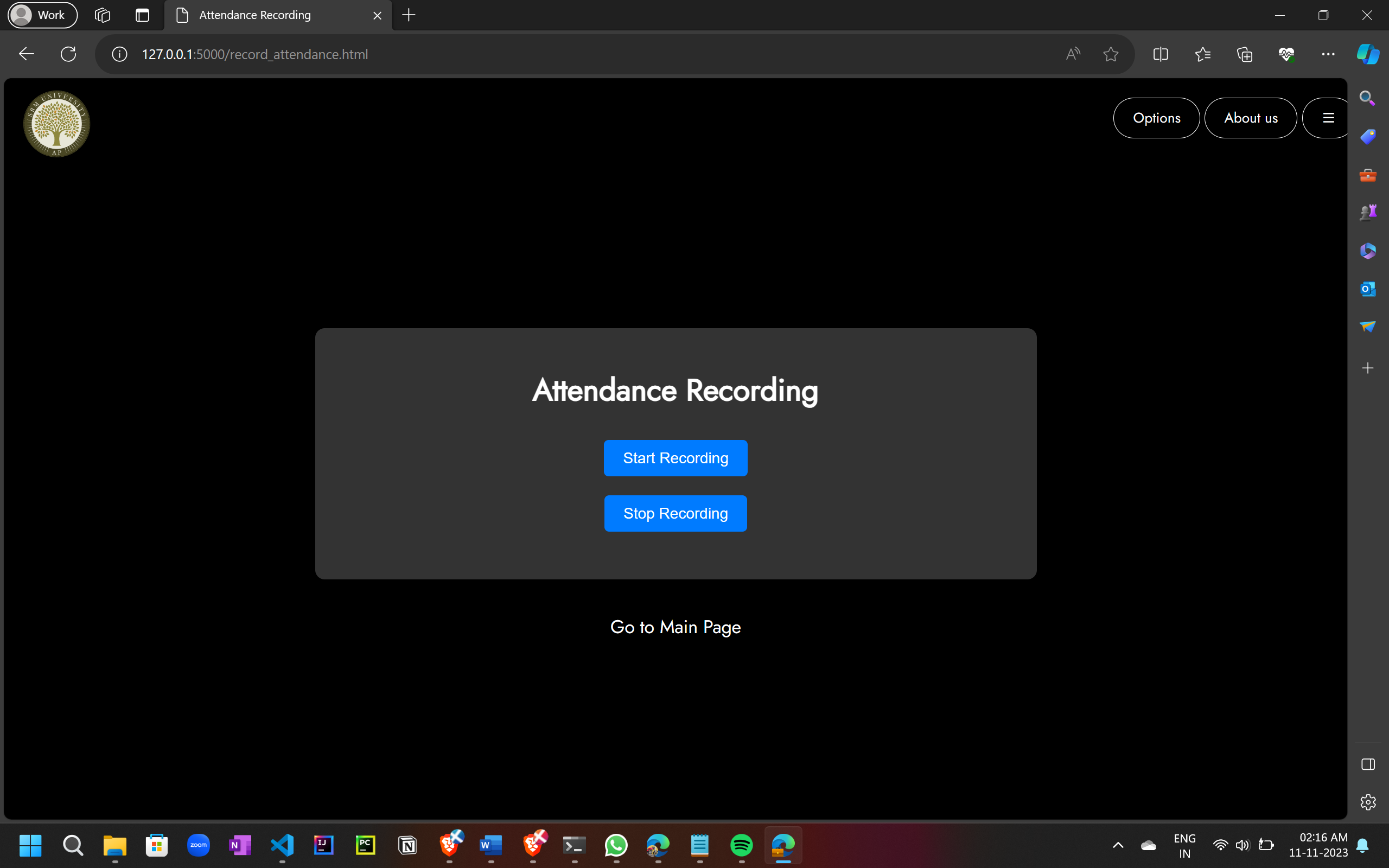
})

**Screenshots**

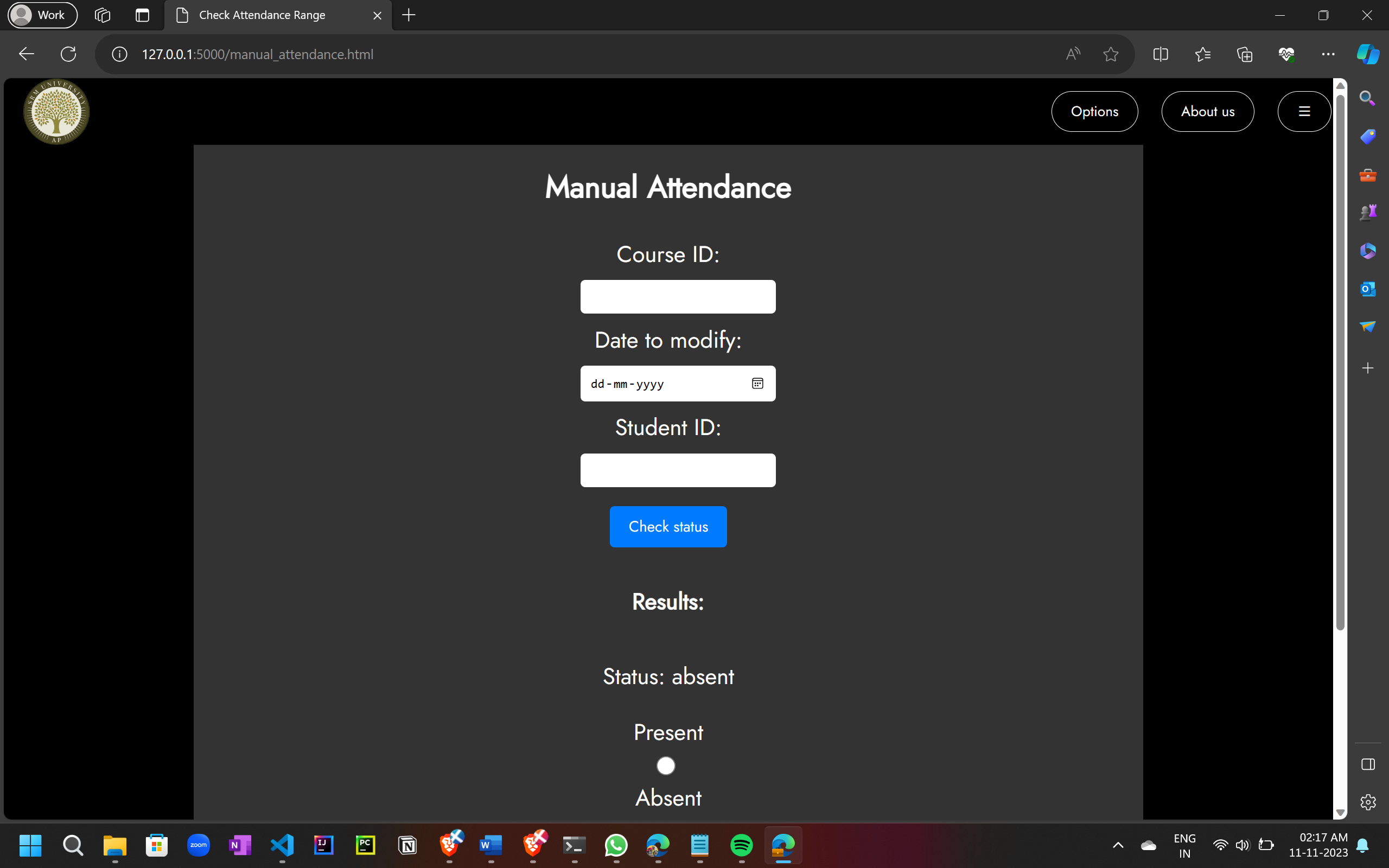
1. **Home page**
2. **Options**



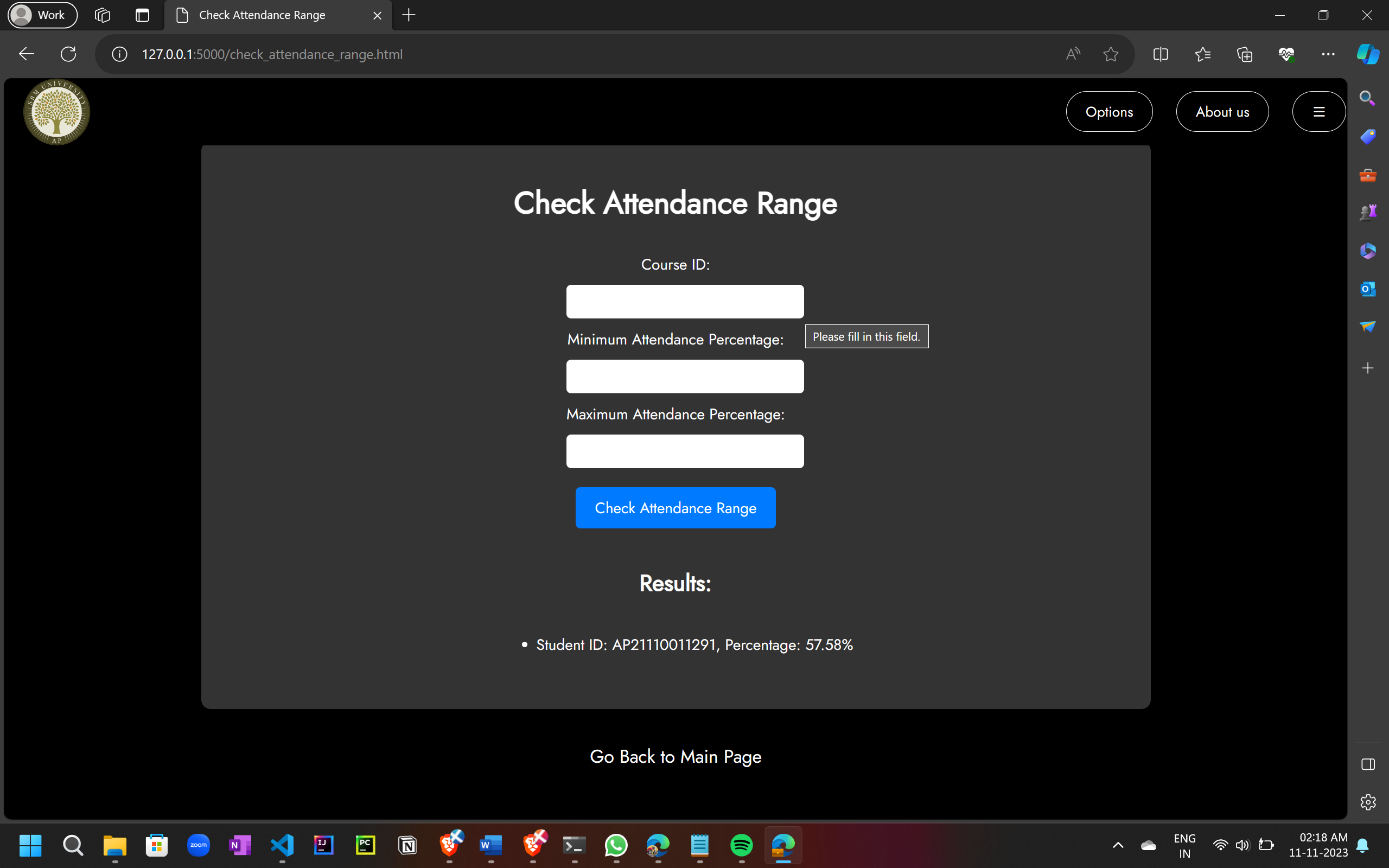
1. **Scanning ID Cards**



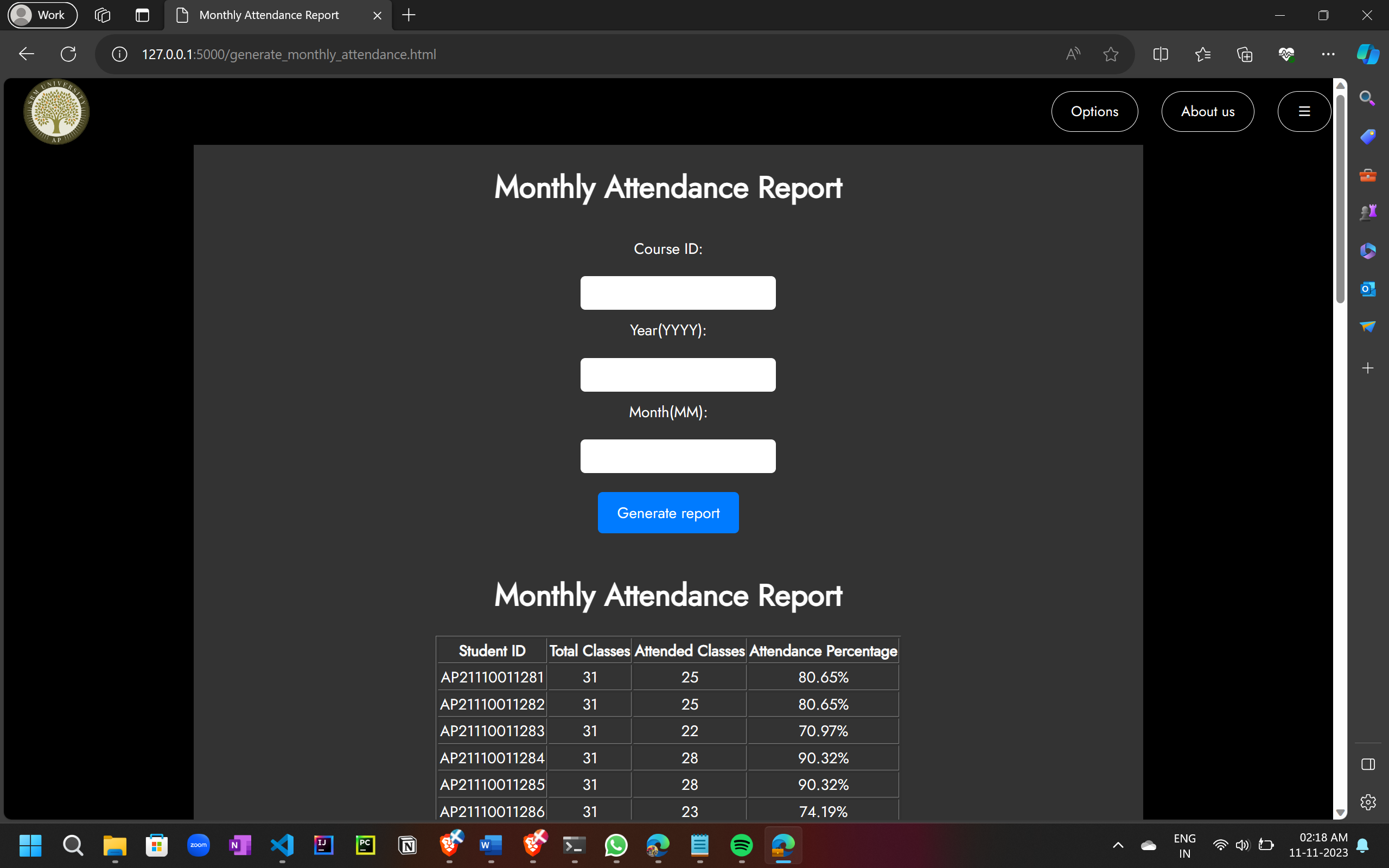
1. **Manual attendance**



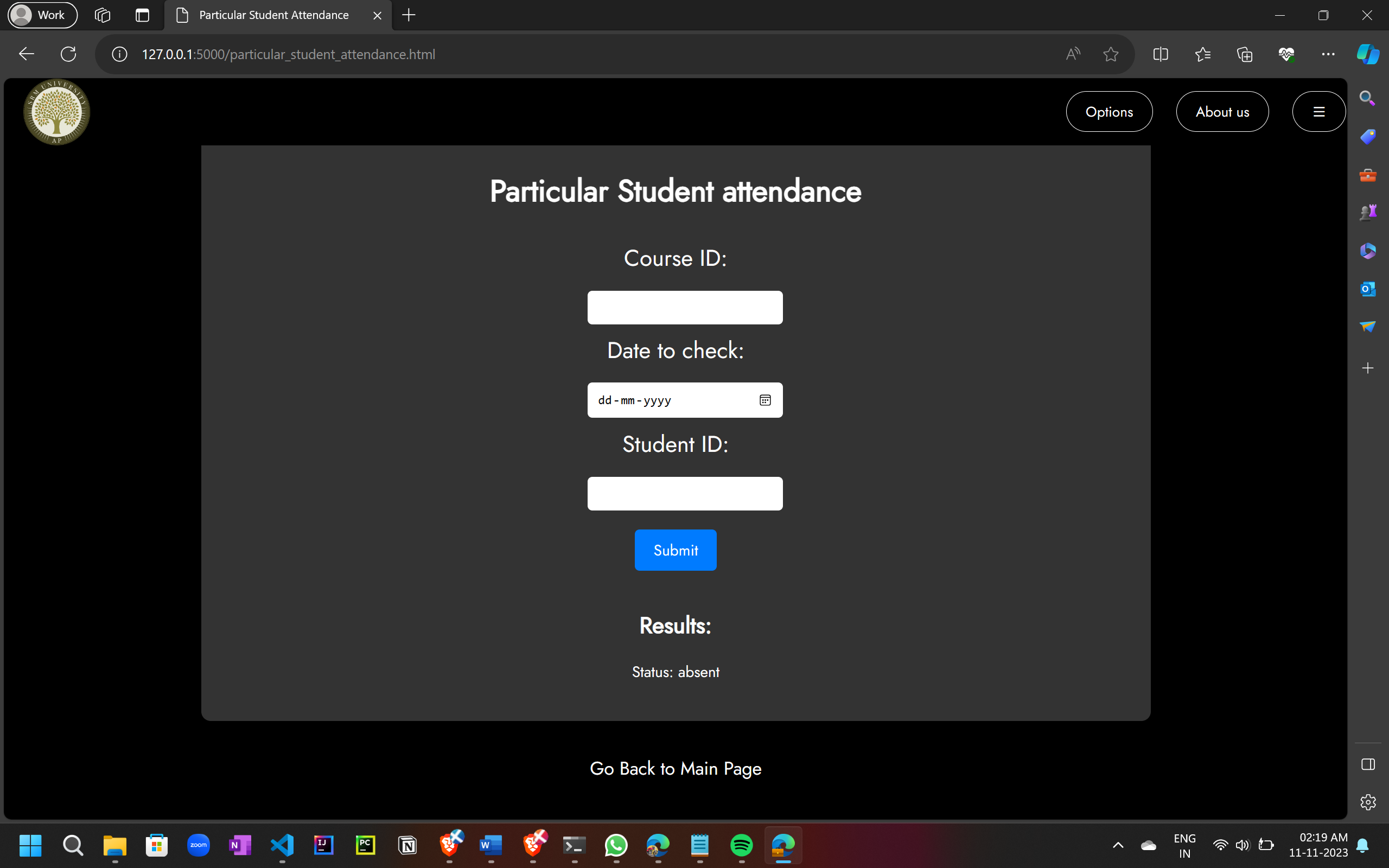
1. **Details of Students within a particular range**



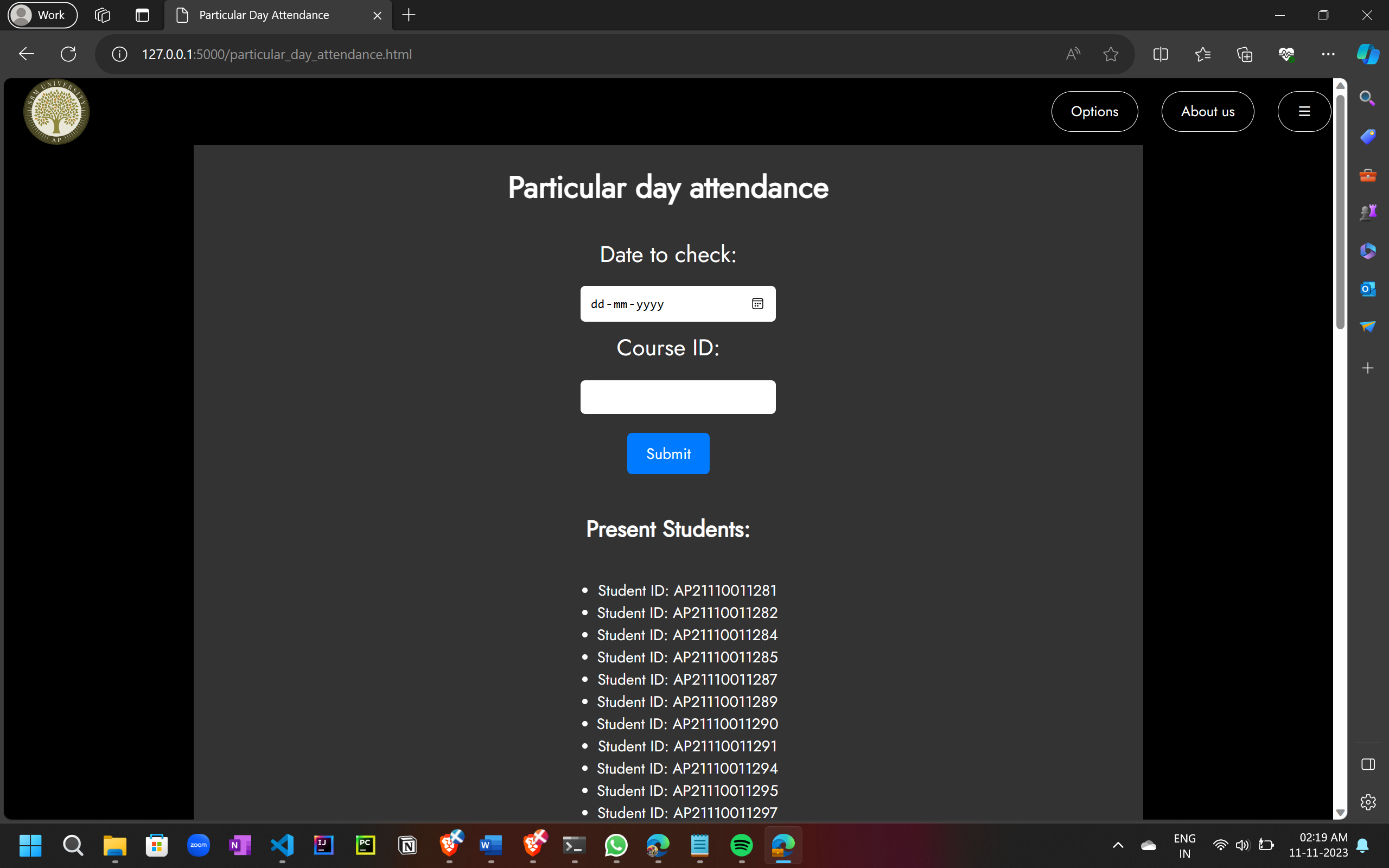
1. **Monthly report**



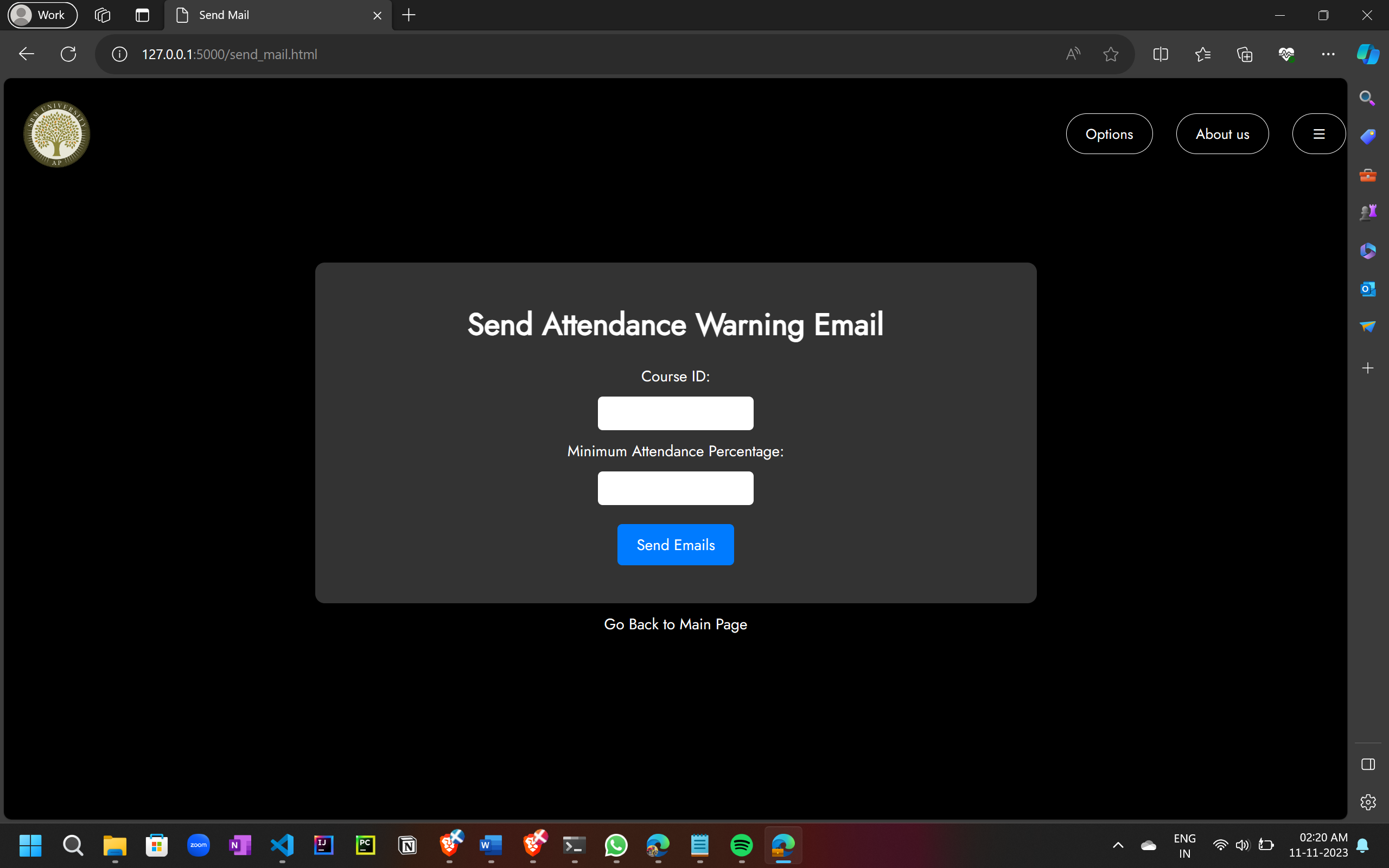
1. **Check the attendance of a student**



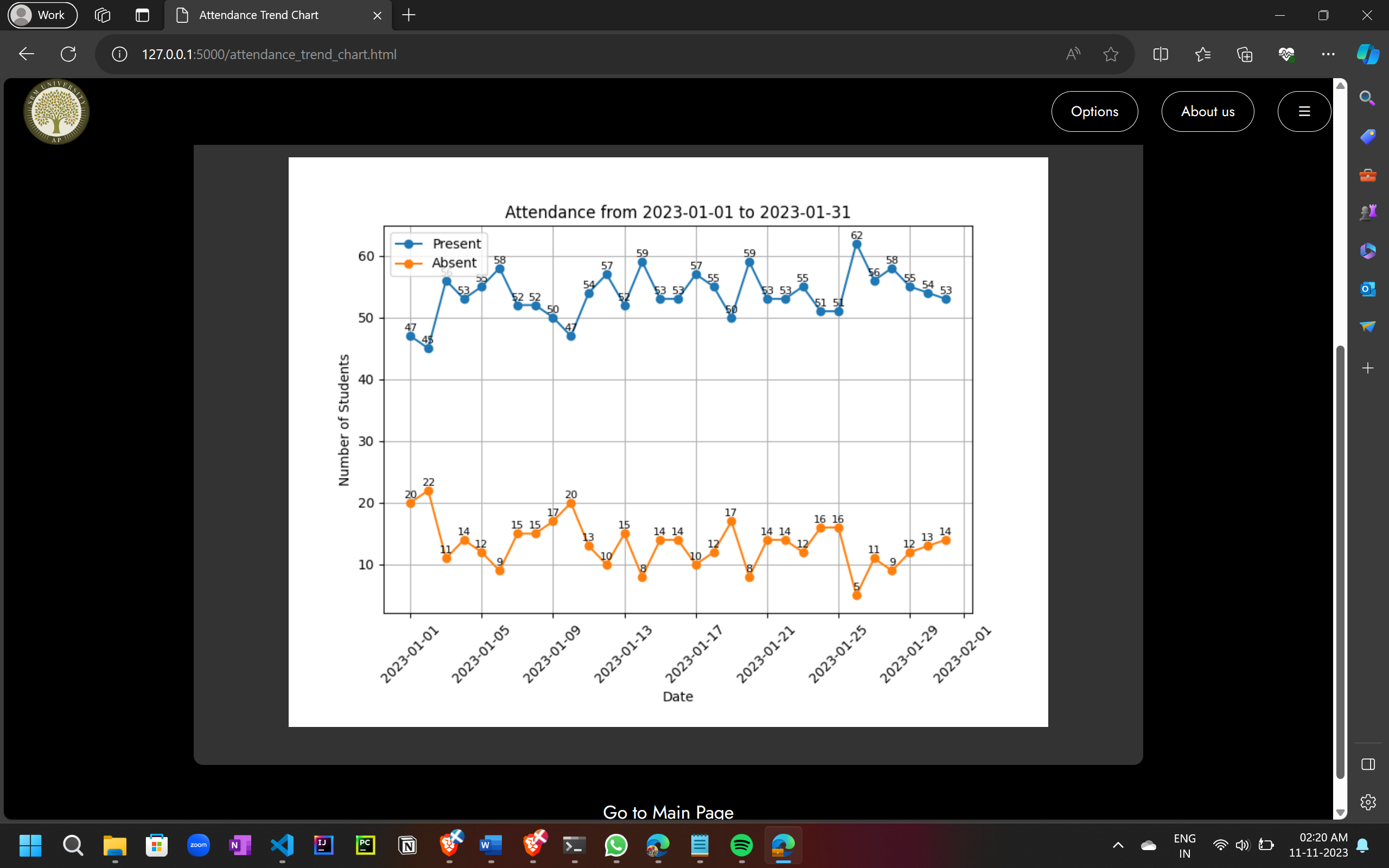
1. **Attendance report of a particular day**



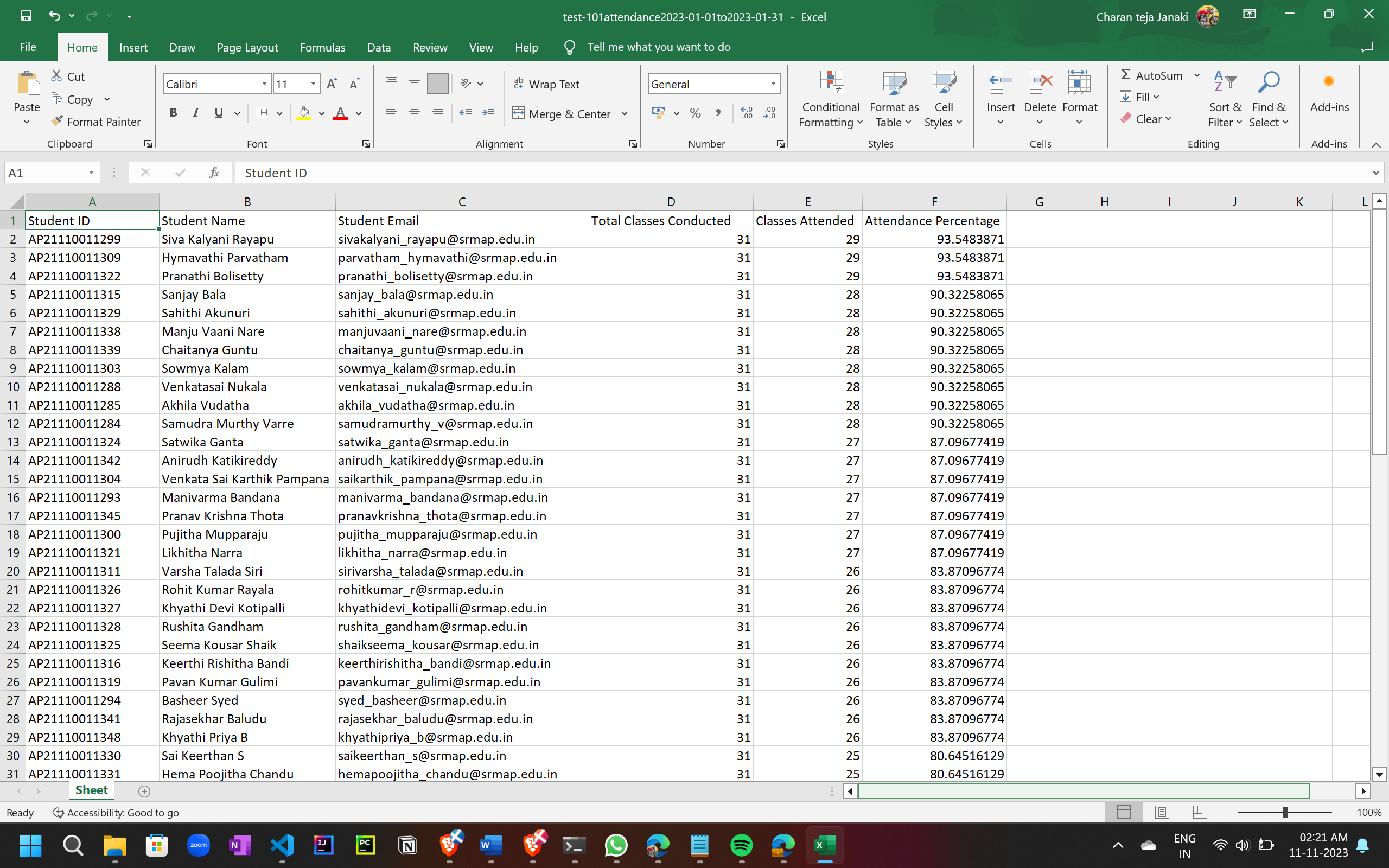
1. **Send alert mails**



1. **Attendance Trend chart**



1. **Get Attendance details in Excel format**



**Conclusion**

The Smart Attendance Management System stands at the forefront of educational innovation, transcending its role as a mere technological solution to become a transformative force in the educational landscape. In essence, it is not just a system; it is a catalyst poised to revolutionize how educational institutions manage attendance, streamline operations, and foster a more engaged and productive learning environment.

By automating the traditionally cumbersome process of attendance tracking, the system liberates educators from administrative burdens, allowing them to redirect their focus toward what truly matters – effective teaching and student engagement. The integration of RFID technology and Flask-powered back-end ensures not only efficiency in data handling but also real-time insights into attendance patterns, facilitating informed decision-making for educators and administrators alike.

Beyond its technical capabilities, the Smart Attendance Management System embodies the essence of educational transformation. It aspires to cultivate an environment where students are not only accounted for but actively supported in their educational journey. The promise lies not just in the reduction of manual workload but in the enhancement of overall educational experiences, contributing to a more enriching and dynamic academic setting.

As the system seamlessly bridges the gap between technology and education, its successful implementation holds the potential to shape a brighter future for educational institutions and their stakeholders. It is not merely about automating processes; it is about creating a paradigm shift towards a more student-centric and efficient educational ecosystem. The Smart Attendance Management System is poised to be an integral part of this evolution, setting the stage for a new era in educational administration and student engagement.

**Future Scope**

This project involves several areas for potential improvement and expansion:

* Enhancing the user interface for a more interactive and user-friendly experience.
* Adding security features to protect sensitive attendance data.
* Implementing user authentication and role-based access control.
* Developing a mobile application for remote attendance tracking.
* Integrating data analytics for generating insightful reports.
* Modifying the schema of the database to provide appropriate views for the students

**References**

1. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.
2. Finkenzeller, K. (2010). RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification, and Near-Field Communication. John Wiley & Sons.
3. Dubois, P. (2008). MySQL Cookbook: Solutions for Database Developers and Administrators. O'Reilly Media.
4. Ramakrishnan, R., & Gehrke, J. (2003). Database Management Systems. McGraw-Hill.
5. Locomotive Scroll. (n.d.). GitHub Repository. Retrieved from <https://github.com/locomotivemtl/locomotive-scroll>
6. Date, C. J. (2003). An Introduction to Database Systems. Addison-Wesley.
7. Chodorow, K., & Dirolf, M. (2010). MongoDB: The Definitive Guide. O'Reilly Media.
8. Coronel, C., Morris, S., & Rob, P. (2016). Database Systems: Design, Implementation, & Management. Cengage Learning.