**DEMONSTRATION OF PROPOSED FEATURES**

# app.py

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

from flask\_sqlalchemy import SQLAlchemy

app = Flask(\_name\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///candidates.db'

db = SQLAlchemy(app)

class Candidate(db.Model):

id = db.Column(db.Integer, primary\_key=True)

name = db.Column(db.String(50), nullable=False)

internal\_marks = db.Column(db.Integer)

communication\_logs = db.relationship('CommunicationLog', backref='candidate', lazy=True)

class CommunicationLog(db.Model):

id = db.Column(db.Integer, primary\_key=True)

timestamp = db.Column(db.String(20))

content = db.Column(db.String(200))

candidate\_id = db.Column(db.Integer, db.ForeignKey('candidate.id'), nullable=False)

db.create\_all()

@app.route('/')

def index():

candidates = Candidate.query.all()

return render\_template('index.html', candidates=candidates)

@app.route('/add\_candidate', methods=['POST'])

def add\_candidate():

if request.method == 'POST':

name = request.form['name']

internal\_marks = request.form['internal\_marks']

new\_candidate = Candidate(name=name, internal\_marks=internal\_marks)

db.session.add(new\_candidate)

db.session.commit()

return redirect(url\_for('index'))

@app.route('/view\_candidate/<int:id>')

def view\_candidate(id):

candidate = Candidate.query.get(id)

return render\_template('view\_candidate.html', candidate=candidate)

if \_name\_ == '\_main\_':

app.run(debug=True)