REFERENCES

Ahvenniemi, H., Huovila, A., Pinto- Seppä, I. And Airaksinen, M., "What are the differences between sustainable and smart cities?," *Cities* **60** (2017) 234–245.

Ameen, R.F.M. and Mourshed, M., "Urban sustainability assessment framework development: The ranking and weighting of sustainability indicators using analytic hierarchy process," *Sustainable Cities and Society*,**44** (2019) 356-366.

AMRUT, "Bihar State Annual Action Plan for proposed schemes under AMRUT," Urban Development & Housing Department, 2015, Accessed 15 March 2017 http://amrut.gov.in/upload/uploadfiles/files/19%20BiharSAAP.pdf .

Bibri, S.E., "The IoT for smart sustainable cities of the future: An analytical framework for sensor-based big data applications for environmental sustainability," *Sustainable Cities and Society*, **38** (2018**)** 230-253.

Bibri, SE. and Krogstie, J., "Smart Sustainable Cities of the Future: An Extensive Interdisciplinary Literature Review," *Sustainable Cities and Society*. **31** (2017) 183-212.

BIS, "*Smart cities - indicators lCS 13.020.20,"* New Delhi: Bureau of Indian Standards, Smart Cities Sectional Committee, 2016.

Booty, W., and Wong, I., "Case Studies of Canadian Environmental Decision Support Systems," *Decision Support Systems*, 2010. Available from: http://www.intechopen.com/articles/show/title/case-studies-of-canadian-environmental-decision-support-systems.

Bosch, P., Jongeneel, S., Rovers, V., Neumann, H.M., Airaksinen, M., and Huovila, A., "CITYkeys indicators for smart city projects and smart cities," CITYkeys report, 2017.

Bruke, S., "Decision-support systems for managing water resources," Environmental Modelling Finding Simplicity in Complexity, Eds. Wainwright, J. and Mulligan, M., John Wiley & Sons Ltd, UK, 2004.

Bulkeley, H. And Betsill, M., "Rethinking sustainable cities: Multilevel governance and the urban politics of climate change," *Environmental Politics*, **14** (2005) 42–63.

Campbell, S., “Green cities, growing cities, just cities? Urban planning and the contradictions of sustainable development,” *Journal of the American Planning Association,* **62** (1996) 296-312.

Cortes, U., Sanchez-Marre, M., Sanguesa, R., Comas, J., Roda, R.I., Poch, M., and Riano, D., “Knowledge management in environmental decision support systems,” *AI Communications*, **14** (1) (2001) 3–12.

Dameri, R., and Cocchia, A, “Smart city and digital city: Twenty years of terminology evolution,”. X conference of the italian chapter of AIS, ITAIS, 2013 pp 18.

Denzer, R, “Generic integration of environmental decision support systems – state-of-the-art,” *Environmental Modelling and Software*, **20** (10) (2005) 1217-1223.

GOI, “Draft Concept Note on Smart City Scheme,” *New Delhi: Ministry of Urban* Development, Government of India, 2014.

Guariso, G., and Werthner, H. (1989). Environmental Decision Support Systems, John Wiley & Sons, ISBN:0-470-21431-7

Haagsma, I.G. and Johanns, R.D., “Decision support systems: An integrated and distributed approach,” *Computational Mechanics,* (1994) 205-212.

Höjer, M., and Wangel, S., “Smart sustainable cities: Definition and challenges,” In L. Hilty, & B. Aebischer (Eds.), ICT innovations for sustainability. Berlin: Springer-verlag, 2015 pp. 333–349.

Khadke, P.A. and Waghmare, P.B, “Class and Size wise Distribution of Urban Centre and their Determinants in Maharashtra,” *RESEARCH REVIEW International Journal of Multidisciplinary*, **3** (2018) 722-730.

Maheshwari, B., Singh, V.P. and Thoradeniya, B., “Balanced urban development: options and strategies for liveable cities,” Springer Nature, (2016) 601.

Marsal-Llacuna, M.L., Colomer-Llinàs, J. and Meléndez-Frigola, J., “Lessons in urban monitoring taken from sustainable and livable cities to better address the Smart Cities initiative,” *Technological Forecasting and Social Change*, **90** (2015) 611-622.

Maslow, A.H., “A theory of human motivation,” *Psychological review*, **50**(4) (1943) 370.

Matthiesa, M., Giupponib, C., and Ostendorf, B., “Environmental decision support systems: Current issues, methods and tools,” *Environmental Modelling & Software*, **22** (2) (2007)123-127.

MEM, “Cleaner greener and safer Mauritius, environmental guideline for smart cities," Ministry of Environment, Sustainable Development, and Disaster and Beach Management, 2015.

MoHUA, “Heritage City Development and Augmentation Yojana,” Ministry of Housing and Urban Affairs, Government of India, 2017.

MoUD, “Jawahar nehru national urban renewal mission,” Ministry of Urban Employment and Poverty Alleviation and Ministry of Urban Development, Government of India, 2006.

MoUD, “Smart Cities: Mission Statement & Guidelines,” New Delhi: Ministry of Urban Development, Government of India, 2015, Accessed 3 March 2016. http://smartcities.gov.in/upload/uploadfiles/files/SmartCityGuidelines(1).pdf.

Randhawa A. and Kumar A., “Exploring sustainability of smart development initiatives in India,” *International Journal of Sustainable Built Environment,* **6** (2017) 701-710.

Sankhe, S., Vittal, I., Dobbs, R., Mohan, A., Gulati, A., Ablett, J., Gupta, S., Kim, A., Paul, S., Sanghvi, A. and Setyy, G., “India's urban awakening: Building inclusive cities, sustaining economic growth,” Mumbai: McKinsey Global Institute, (2010) 234.

Scott Morton, M.S, “Management Decision Systems; Computer-based support for decision making,” Boston, Division of Research, Graduate School of Business Administration, Harvard Universit*,* 1971.

Singh, P.K., Shruti and Ohri, A, “Selecting Environmental Indicators for Sustainable Smart Cities Mission in India,” , *Nature Environment and Pollution Technology*, **19**(1) (2020) 201-210.

The World Bank data, 2019, Accessed 2 February 2020. https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS?locations=IN.

Turner, BL, The earth as transformed by human action: global and regional changes in the biosphere over the past 300 years: CUP archive, 1990.

WCED, S.W.S., “World commission on environment and development. Our common future” **17** (1987) 1-91.

Yadav, G., Mangla, S.K., Luthra, S. And Rai, D.P., “Developing a sustainable smart city framework for developing economies: An Indian context,” *Sustainable Cities and Society,* **47** (2019) 101462.

Bansal, S., Pandey, V. and Sen, J., “Redefining and exploring the smart city concept in Indian perspective: case study of Varanasi,” In From Poverty, Inequality to Smart City Springer, Singapore, 2017 pp. 93-107.

CBUD, 2015a. City Development Plan for Allahabad, 2041 (Final City Development Plan).Retrievedfromhttp://allahabadmc.gov.in/documentslist/City_Development_Plan_Allahabad-2041.pdf .

CBUD, 2015b. City Development Plan for Varanasi, 2041 (Final City Development Plan). Retrieved from http://nnvns.org/data/Final%20CDP%20Varanasi.pdf.

CPHEEO, “Municipal Solid Waste Management Manual”, 2016.

Dong, Y. and Hauschild, M.Z., “Indicators for Environmental Sustainability,” In Procedia CIRP, 61: 697-702, 2017.

Garau, C. and Pavan, V. M., “Evaluating urban quality: Indicators and assessment tools for smart sustainable cities,” *Sustainability* **10**(3), (2018) 575.

Howard, G. and Bartram, J., “Effective water supply surveillance in urban areas of developing countries,” *Journal of Water Health*, **3** (2005) 31-42.

ITU , “Agreed definition of a smart sustainable city, Focus Group on Smart Sustainable Cities,” SSC-0146 version Geneva, 5-6 March, 2014.

JICA, 2016. Data Collection Survey on Improvement of Environment in Varanasi City, Republic of India Final Report (Main Report).

MoUD, “Service levels in Urban water and sanitation sector Status Report (2010-2011),” Ministry of Urban Development, Government of India, 2012.

Rasadurai, K., Kalanidhi, K. and kumar, V., "Smart City Billing System for Homes through IoT," *International Journal of Engineering Research in Computer Science and Engineering*, **4** (2017) 2394-2320.

Rezaei, J., "Best-worst multi-criteria decision-making method," *Omega*, **53** (2015) 49-57

Unnisa, S.A. and Hassan, M.N., "Study on Water Supply Services of Hyderabad and Warangal ULBs Compared to the Standards of Service Level Benchmarks," *Journal of Management Science and Practice*, **1**(2) (2013) 45.

AMRUT Lucknow, "SLIP OF WATER SUPPLY OF LUCKNOW CITY**,"** 2015.

Anand, A., Rufuss, D.D.W., Rajkumar, V., and Suganthi, L., "Evaluation of sustainability indicators in smart cities for India using MCDM approach," *Energy Procedia***, 141** (2017) 211-215.

Awad-Núñez, S., González-Cancelas, N. and Camarero-Orive, A., "Application of a model based on the use of DELPHI methodology and Multicriteria Analysis for the assessment of the quality of the Spanish Dry Ports location," *Procedia Social and Behavioral Sciences,* **162**, (2014) 42-50.

Bartolozzi, M., Bellini, P., Nesi, P., Pantaleo, G. and Santi, L., "A smart decision support system for smart city," IEEE International Conference on Smart City, 2015 pp 117-122.

CDP Allahabad, "City Development Plan for Allahabad, 2041," Ministry of Urban Development, GOI, 2015.

Cities, S., "Indicators for Sustainability: How cities are monitoring and evaluating their success," 2012.

Cohen, B., "The Smart Cities in the World: Methodology," 2012.

Deng, W., Peng, Z. And Tang, Y.T., "A quick assessment method to evaluate sustainability of urban built environment: Case studies of four large-sized Chinese cities," *Cities* **89** (2019) 57-69.

Egger, S. , "Determining a sustainable city model," *Environmental modelling and software*, **2** (2006)1235–1246.

EIU , "The Green City Index," The Economist Intelligence Unit (EIU), 2012. http://aiph.org/wp-content/uploads/2015/04/Green City - Guidelines.pdf.

Girardet, H., “Creating sustainable cities,” Foxhole, Dartington, Totnes, Devon,UK: Green Books, 1999.

Greenstone, M., Harish, S., Pande, R. and Sudarshan, A., “The solvable challenge of air pollution in India,” In India Policy Forum, 2017 pp. 11-12.

Hofstad, H., “Compact city development: High ideals and emerging practices,” *European Journal of Sustainable Development*, **49** (2012) 1–23.

Huovila, A., Bosch, P. And Airaksinen, M., “Comparative analysis of standardized indicators for Smart sustainable cities:What indicators and standards to use and when?,” *Cities* **89** (2019) 141-153.

ISB, “Smart Cities Index A tool for Evaluating Cities,” 2017.

Jenks, M., Burton, E. and Williams, K. (Eds.), “The compact city: A sustainable urban form?,” In. London: E&FN Spon Press, 1996a.

Jenks, M., Burton, E. And Williams, K., “A sustainable future through the compact city?,” Urban intensification in the United Kingdom, 1996b pp 5–20.

Joshi, S., Saxena, S. And Godbole, T., “Developing smart cities: An integrated framework.” *Procedia Computer Science*, **93** (2016) 902-909.

Joss, S., “Eco-cities – a global survey,” *WIT Transactions on Ecology and Environment*, **129** (2010) 239–250.

Joss, S., “Eco-cities: The mainstreaming of urban sustainability; key characteristics and driving factors,” *International Journal of Sustainable Development and Planning,* **6** (2011) 268–285.

Joss, S., Cowley, R., and Tomozeiu, D., “Towards the ubiquitous eco-city: An analysis of the internationalisation of eco-city policy and practice,” *Journal of urban research and practice,* **76** (2013) 16–22.

Juan, Y.K., Wang, L., Wang, J., Leckie, J.O. and Li, K.M., “A decision-support system for smarter city planning and management,” *IBM Journal of Research and Development*, **55**(1.2) (2011) 3-1.

Komninos, N., “Intelligent Cities and Globalisation of Innovation Networks,” London: Routledge, 2008.

Kumar, S., Luthra, S., Govindan, K., Kumar, N. And Haleem, A., "Barriers in green lean six sigma product development process: an ISM approach," *Production Planning & Control*, **27** (2016) 604-620.

Macharis, C., Springael, J., De Brucker, K. and Verbeke, A., "PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP," *European Journal of Operational Re*search, **153** (2004)307-317.

Miles, A., Zaslavsky, A., and Browne, C., " IoT-based Decision Support System for Monitoring and Mitigating Atmospheric Pollution in Smart Cities," *Journal of Decision Systems,* **27** (2018) 56–67.

MoEF, "Air and noise pollution in Delhi and NCR", 2017.

MoUD, "Swachh Survekshan A Guidebook for Urban Local Bodies," 2017. http://164.100.228.143:8080/sbm/content/writereaddata/SS_GuideBook.pdf.

OECD, "Environmental Indicators Towards Sustainable Development," 2001. https://www.oecd.org/site/worldforum/33703867.pdf

OMC-KORBA, "Template For Service Level Improvement Plan Sewerage And Septage," Office of Municipal Copporation, 2017.

Rana, N. P., Luthra, S., Mangla, S. K., Islam, R., Roderick, S.,and Dwivedi, Y. K., "Barriers to the development of smart cities in Indian context," *Information Systems Frontiers*, **21**(3) (2018) 503-525.

Science for Environment Policy—European Commission, "Indicators for sustainable cities depth Report," **12** (2015).

Segnestam, L., "Indicators of Environment and Sustainable Development Theories and Practical Experience," *ENVIRONMENTAL ECONOMICS SERIES*, **89** (2002).

Sureshchandra, M.S., Bhavsar,J.J. and Pitroda R.J, "Assessment Of Critical Success Factors for Smart Cities Using Significance Index Method," *International Journal of Advance Research and Innovative Ideas in Education*. **2**, (2016) 802-810.

Tare, V. and Bose, P., “Compendium of Sewage Treatment Technologies,” National River Conservation Directorate, Ministry of Environment and Forests, Government of India, 2009.

UCI, “The China Urban Sustainability Index (USI) 2013,”, authored by Xiaopeng Li, Xiujun Li, Jonathan Woetzel, Gengtian Zhang and Yingjie Zhang, The Urban China Initiative, 2014 pp1-45.

UPPCB , 2017b. http://www.uppcb.com/air_quality_april.html.

UPPCB, ENVIS Centre : Uttar Pradesh, 2017a.

Venkatesh., G., “A critique of the European Green City Index,” *Journal of Environmental Planning and Management,* **57** (2014) 317-328.

Williams, K., “Sustainable cities: Research and practice challenges*,” International Journal of Urban Sustainable Development,* (2009) 128–132.

WSP, “The Karnataka Urban Water Sector Improvement Project,” 2010

Yadav, G. and Desai, T.N., “A fuzzy AHP approach to prioritize the barriers of integrated Lean Six Sigma,” *International Journal of Quality & Reliability Management*, **34** (2017)1167-1185.

Kumar, D. and Alappat, BJ., “Errors Involved in the Estimation of Leachate Pollution Index,” *Practice Periodical of Hazardous, Toxic, and Radioactive Waste Management*, **2** (2005) 103-111.

Sözer, H. and Takmaz, D., “Calculation of the Sensitivity Factors within the Defined Indexes in a Building Level,” *Journal of Sustainable Development of Energy, Water and Environment Systems* ., **1** (2020) 1-21.

Saisana, M. and Saltelli, A., “Sensitivity Analysis for the 2008 Environmental Performance Index,” European Commission: Fermi, 2008.

Pianosi, F., Beven, K., Freer, J., Hall, JW., Rougier, J., Stephenson, DB. and Wagener, T., “Sensitivity analysis of environmental models: A systematic review with practical workflow,” *Environmental Modelling Software*, **79** (2016) 214-232.

Ohri, A. and Singh, PK., “Error Involved in Estimation of Site Sensitivity Index (SSI) for Landfilling of Municipal Solid Waste,” *International Journal of Environmental Science*, **1** (2011) 767-770.

LIST OF PUBLICATIONS

1) Shruti, Singh, P.K., and Ohri, A (2021). Sensitivity Analysis of the Smart City Environmental Sustainability Index (SCESI). Nature Environment and Pollution Technology. 20(2), pp 703-711.

2) Shruti, Singh, P.K., and Ohri, A (2021). Evaluating the Environmental Sustainability of Smart Cities in India: The Design and Application of the Indian Smart City Environmental Sustainability Index. Sustainability,13, pp 327.

3) Singh, P.K., Shruti, and Ohri, A (2020). Selecting Environmental Indicators for Sustainable Smart Cities Mission in India. Nature Environment and Pollution Technology,19(1), pp 201-210.

4) Decision Support Framework for the Assessment of Environmentally Sustainable Smart Cities in India. (Environmental Progress and Sustainable Energy- Revision submitted).

5) Multi-criteria Decision-Making for Environmentally Sustainable Smart Cities Assessment. (Environment, Development and Sustainability- Under Review).

Appendix A

Program codes for DSS-ESSC

```
from django.shortcuts import render,HttpResponseRedirect,render_to_response
from django.template import RequestContext
#from .models import Question
from django.conf import settings
from django.contrib.auth.models import User
from django.contrib.auth import authenticate
from django.contrib.auth import logout
from django.contrib import messages
from django.core.mail import send_mail
from django.db import models,connection
from django.contrib.auth.decorators import login_required
from django.contrib.auth import login as auth_login
from django.db.models import F
from django.db.models.expressions import CombinedExpression, Value
from django.shortcuts import redirect
from django.core.mail import send_mail
import MySQLdb
from django.core.files.storage import FileSystemStorage
from smartcity.settings import MEDIA_ROOT
from math import exp,log
# Create your views here.
from django.shortcuts import render
from django.shortcuts import render,redirect
from django.db import connection
from django.http import HttpResponse
import MySQLdb
from django.contrib.auth.models import User
```

```
from django.contrib import auth
import datetime
from django.contrib.auth.decorators import login_required
#import requests
import random


def critical_rating(request):
        s=""
        c = connection.cursor()
        if request.method=="GET":
                return render(request,'critical_rating.html')


        if request.method == 'POST':
                parameter = str(request.POST.get('parameter'))
                city = str(request.POST.get('city'))
                c.execute("select "+parameter+" from garbage where city='%s'"%(city))
                row=c.fetchone()
                x=float(row[0])


                if(x<50):
                        s="Highly Critical"
                elif(x<75):
                        s="Moderately Critical"
                else:
                        s="Lowly critical"
                x=x/100
                return render(request,'critical_rating.html',{'s':s,'x':x})


def population_prediction(request):
```

```
        c = connection.cursor()
        if request.method=="GET":
                return render(request,'population_prediction.html')

        if request.method == 'POST':
                year = int(request.POST.get('year'))
                city = str(request.POST.get('city'))
                c.execute("select "+city+" from population ")
                row=c.fetchall()
                l=[]
                for i in row:
                   l.append(i[0])
                thres_rate=0.05
                if year==0:
                future_pop=l[-1]
            else:
                   r=log(l[-1]/float(l[-min(year+1,len(l))]))
                   r=r/min(year,len(l)-1);
                   print r
                   if r>thres_rate:
                         r=thres_rate
                   future_pop=l[-1]*exp(r*year)
                   print int(future_pop)
                return
render(request,'population_prediction.html',{"pop":int(future_pop)})

def environment_index(request):
        if request.method=="GET":
                return render(request,'environment_index.html')
        if request.method == 'POST':
```

```
                type = str(request.POST.get('type'))
                c = connection.cursor()
                c.execute("select parameter, weight from env_index where type
='%s'"%(type))
                row=c.fetchall()
                l=[]
                w=[]
                new=[]
                for i in row:
                    l=[]
                    l.append(i[0])
                    l.append(float(i[1]))
                    new.append(l)
                return render(request,'calculate_index.html',{"l":new,"type":type})



def calculate_index(request):
        if request.method=="GET":
                return render(request,'environment_index.html')
        if request.method == 'POST':
                type = str(request.POST.get('type'))
                c = connection.cursor()
                c.execute("select parameter, weight from env_index where type
='%s'"%(type))
                row=c.fetchall()
                l=[]
                new=[]
                sum=0
                sum1=0
                for i in row:
```

```
                    l=[]
                    l.append(i[0])
                    l.append(float(i[1]))
                    x= float(str(request.POST.get((i[0]))))
                    l.append(x*float(i[1]))
                    l1=[]
                    l1.append(type)
                    l1.append(i[0])
                    l1.append(float(i[1]))
                    y=f(l1)
                    l.append(y)
                    sum+=x*float(i[1])
                    sum1+=x
                    new.append(l)
                tot=""
                sum/=sum1
                if(sum<0.3):
                    tot="Highly Critical"
                elif(sum<0.6):
                    tot="Moderately Critical"
                elif(sum<0.9):
                    tot="Low Critical"
                else:
                    tot="Not Critical"
                return
render(request,'final_result.html',{"type":type,"sum":sum,"l":new,"tot":tot})



def review(request):
        if request.method=="GET":
```

```
                return render(request,'review.html')
        if request.method == 'POST':
                #comment = str(request.POST.get('comment'))
                #sentiment = getsenti.test(comment)
                #print sentiment
                c = connection.cursor()
                c.execute("select parameter, weight from env_index where type
='%s'"%(type))
                return
render(request,'review.html',{"type":type,"sum":sum,"l":new,"tot":tot})


def home(request):
        if request.method=="GET":
                return render(request,'city.html')



def index(request):
    return render(request,'index2.html')


def welcome(request):
    return render(request,'index.htm')



def action_plan(request):
    return render(request,'action.html')



def solid_waste_mangement(request):
    type = 'Solid_waste_management'
    return render(request,'solid.html',{"type":type})
```

```
def water_supply(request):
    type = 'Water_Supply'
    return render(request,'water.html',{"type":type})

# sewerage and sanitation + storm water drainage
def sewerage_and_sanitation(request):
    type = 'Sewerage_and_sanitation'
    return render(request,'sewerage.html',{"type":type})

def ambient_environment(request):
    type = 'Ambient_Environment'
    return render(request,'environment.html',{"type":type})

def index_range(score):
    r = ""
    if score<=20:
        r = "Critically low"
    elif score<=40:
        r = "Poor"
    elif score<=60:
        r = "Fair"
    elif score<=80:
        r = "Good"
    elif score<=100:
        r = "Excellent"
    return r

def top_three(dictonary,sorted_list):
```

```
    top = []
    i=0
    for s in sorted_list:
        f = False
        for d in dictonary:
            if s==d['value']:
                top.append(d)
                i+=1
            if i==3:
                f = True
                break
        if f:
            break
    return top



def get_table(column,data):
    table_list=[]
    i=2
    while(i<len(column)):
        table_list.append({'name':column[i],'value':data[i]})
        i+=1
    return table_list;

# this function convert tuple to dictonary
def convert_to_dict(rows):
    d = {}
    for row in rows:
        d[row[0].lower()] = row[1]
```

```
    return d


def get_top_three(sorted_list,weights):
    swm = sorted_list[:3]
    for index in range(3):
        swap = index
        for i in range(index+1,3):
            if swm[index][1]==swm[i][1]:
                if weights[swm[index][0]]<weights[swm[i][0]]:
                    swap = i
            #print("Need To swap")
        swm[index],swm[swap] = swm[swap],swm[index]


    return swm


def show_result(request):
    #get the last city inserted
    con = connection.cursor()
    con.execute("select name from cities order by(id) desc limit 1")
    rs = con.fetchone()
    city = rs[0]


    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM SWM_Factor WHERE city='"+city+"' order by(id)
desc")
    c.execute("select lower(parameter), weight from env_index where type
='%s'"%('Solid_waste_management'))
    row=c.fetchall()
    rs = con.fetchone()
```

```
    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))
    test =sorted(t.iteritems(), key = lambda x : x[1])
    dict_weights = convert_to_dict(row)
    swm_table_top = get_top_three(test,dict_weights)
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    swm_score = sum(final)*100
    swm_range_value = index_range(swm_score)
    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM WS_Factor WHERE city='"+city+"' order by(id)
desc")
    c.execute("select parameter, weight from env_index where type
='%s'"%('Water_Supply'))
    row=c.fetchall()
    rs = con.fetchone()


    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))
    test =sorted(t.iteritems(), key = lambda x : x[1])
    dict_weights = convert_to_dict(row)
    ws_table_top = get_top_three(test,dict_weights)
```

```
    #ws_table = get_table(map(lambda x:x[0], con.description),rs)
    #rs_sorted = list(rs)
    #rs_sorted.sort()
    #ws_table_top = top_three(ws_table,rs_sorted[:3])
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    ws_score = sum(final)*100
    ws_range_value = index_range(ws_score)


    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM SS_Factor WHERE city='"+city+"' order by(id)
desc")
    c.execute("select parameter, weight from env_index where type
='%s'"%('Sewerage_and_sanitation'))
    row=c.fetchall()
    rs = con.fetchone()


    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))
    test =sorted(t.iteritems(), key = lambda x : x[1])
    dict_weights = convert_to_dict(row)
```

```
    ss_table_top = get_top_three(test,dict_weights)


    #ss_table = get_table(map(lambda x:x[0], con.description),rs)
    #rs_sorted = list(rs)
    #rs_sorted.sort()
    #ss_table_top = top_three(ss_table,rs_sorted[:3])
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    ss_score = sum(final)*100
    ss_range_value = index_range(ss_score)



    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM AE_Factor WHERE city='"+city+"' order by(id)
desc")
    c.execute("select parameter, weight from env_index where type
='%s'"%('Ambient_Environment'))
    row=c.fetchall()
    rs = con.fetchone()


    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))
```

```
test =sorted(t.iteritems(), key = lambda x : x[1])
dict_weights = convert_to_dict(row)
ae_table_top = get_top_three(test,dict_weights)

#ae_table = get_table(map(lambda x:x[0], con.description),rs)
#rs_sorted = list(rs)
#rs_sorted.sort()
#ae_table_top = top_three(ae_table,rs_sorted[:3])
c.close()
con.close()
fac = []
for fval in row:
    fac.append(fval[1])
final=[]
m = 2
for x in fac:
    final.append(float(rs[m])*float(x))
    m+=1
ae_score = sum(final)*100
ae_range_value = index_range(ae_score)

sscei_score = (ae_score+ss_score+ws_score+swm_score)/4

sscei_range = ""
if sscei_score<=20:
    sscei_range = "Critically Low"
elif sscei_score<=40:
    sscei_range = "Poor"
elif sscei_score<=60:
```

```
        sscei_range = "Average"
    elif sscei_score<=80:
        sscei_range = "Very Good"
    elif sscei_score<=100:
        sscei_range = "Excellent"


    temp = [
            {'name':'Solid Waste Management
Index','value':swm_score,'range':swm_range_value,'one':{'name':solid_labels(swm_table_
top[0][0]),'value':swm_table_top[0][1]},'two':{'name':solid_labels(swm_table_top[1][0]),'
value':swm_table_top[1][1]},'three':{'name':solid_labels
(swm_table_top[2][0]),'value':swm_table_top[2][1]}},

            {'name':'Water Supply
Index','value':ws_score,'range':ws_range_value,'one':{'name':water_labels(ws_table_top[0
][0]),'value':ws_table_top[0][1]},'two':{'name':water_labels(ws_table_top[1][0]),'value':w
s_table_top[1][1]},'three':{'name':water_labels(ws_table_top[2][0]),'value':ws_table_top[
2][1]}},

            {'name':'Sewerage Sanitation and Storm Water Drainage
Index','value':ss_score,'range':ss_range_value,'one':{'name':sewerage_labels(ss_table_top[
0][0]),'value':ss_table_top[0][1]},'two':{'name':sewerage_labels(ss_table_top[1][0]),'value
':ss_table_top[1][1]},'three':{'name':sewerage_labels(ss_table_top[2][0]),'value':ss_table_t
op[2][1]}},

            {'name':'Ambient Environment Conditions
Index','value':ae_score,'range':ae_range_value,'one':{'name':ambient_labels(ae_table_top[
0][0]),'value':ae_table_top[0][1]},'two':{'name':ambient_labels(ae_table_top[1][0]),'value'
:ae_table_top[1][1]},'three':{'name':ambient_labels(ae_table_top[2][0]),'value':ae_table_t
op[2][1]}}

            ]

    score_list = [swm_score,ws_score,ss_score,ae_score]

    score_list.sort()

    data = []

    for score in score_list:

        for t in temp:

            if score==t['value']:

                data.append(t)
```

```
    return
render(request,'result.html',{'city':city,'test':test,'data':data,'sscei_val':sscei_score,'sscei_ra
nge':sscei_range})

    #return
render(request,'result.html',{'data':data,'sscei_val':sscei_score,'sscei_range':sscei_range})
```

```
def view_result(request):

    city = str(request.POST.get('city'))

    c = connection.cursor()

    con = connection.cursor()

    con.execute("SELECT * FROM SWM_Factor WHERE city="+city+" order by(id)
desc")

    c.execute("select parameter, weight from env_index where type
='%s'"%('Solid_waste_management'))

    row=c.fetchall()

    rs = con.fetchone()

    c.close()

    con.close()

    fac = []

    for fval in row:

        fac.append(fval[1])

    final=[]

    m = 2

    for x in fac:

        final.append(float(rs[m])*float(x))

        m+=1

    swm_score = sum(final)*100

    swm_range_value = index_range(swm_score)
```

```
    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM WS_Factor WHERE city="+city+" order by(id)
desc")
    c.execute("select parameter, weight from env_index where type
='%s'"%('Water_Supply'))
    row=c.fetchall()
    rs = con.fetchone()
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    ws_score = sum(final)*100
    ws_range_value = index_range(ws_score)


    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM SS_Factor WHERE city="+city+" order by(id) desc")
    c.execute("select parameter, weight from env_index where type
='%s'"%('Sewerage_and_sanitation'))
    row=c.fetchall()
    rs = con.fetchone()
    c.close()
    con.close()
    fac = []
```

```
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
         #changes made
    ss_score = sum(final)*100
    ss_range_value = index_range(ss_score)



    c = connection.cursor()
    con = connection.cursor()
    con.execute("SELECT * FROM EC_Factor WHERE city="+city+" order by(id) desc")
    c.execute("select parameter, weight from env_index where type
='%s'"%('Environment'))
    row=c.fetchall()
    rs = con.fetchone()
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
         #changes made
```

```
    ae_score = sum(final)*100
    # ae_score = sum(final)*25

    ae_range_value = index_range(ae_score)

    #changes made
    sscei_score = (ae_score+ss_score+ws_score+swm_score)/4

    sscei_range = ""
    if sscei_score<=20:
        sscei_range = "Critically Low"
    elif sscei_score<=40:
        sscei_range = "Poor"
    elif sscei_score<=60:
        sscei_range = "Fair"
    elif sscei_score<=80:
        sscei_range = "Good"
    elif sscei_score<=100:
        sscei_range = "Excellent"
    return
render(request,'result.html',{'SWM_val':swm_score,'SWM_range':swm_range_value,'WS
_val':ws_score,'WS_range':ws_range_value,'SS_val':ss_score,'SS_range':ss_range_value,'
EC_val':ec_score,'EC_range':ec_range_value,'sscei_val':sscei_score,'sscei_range':sscei_ra
nge})



def solid_labels(pname):
    if pname == "degree_of_segregation":
        return "Degree of Segregation"
    elif pname == "collection_efficiency":
        return "Efficiency in collection of MSW"
```

```
    elif pname == "msw_recovery":
        return "Extent of Solid Waste Recovered"
    elif pname == "scientific_disposal":
        return "Degree of Scientific Disposal of MSW"
    elif pname =="swmp":
        return "SWM programs carried in the city during 3 years"
    elif pname == "c_and_d":
        return "Availability of separate system for recycling Construction & Demolition
waste"
    elif pname =="cost_recovery":
        return "Extent of cost recovery in SWM"
    return pname


def water_labels(paname):
    if paname == "per_capita_supply":
        return "Adequacy of water supply"
    elif paname == "continuity":
        return "Continuity of water supplied in hrs per day"
    elif paname == "consumption_metering":
        return "Smart meters and management"
    elif paname == "non_revenue_water":
        return "Leakage identification"
    elif paname =="quality":
        return "Water quality"
    elif paname == "cost_recovery":
        return "Extent of cost recovery in WSM"
    return paname


def sewerage_labels(parname):
    if parname == "collection_efficiency":
```

```
        return "Collection efficiency of Sewage network"
    elif parname == "reuse_and_recycling":
        return "wastewater recycling"
    elif parname == "cost_recovery":
        return "Extent of cost recovery in SSS"
    elif parname == "coverage":
        return "Coverage of Storm Water Drainage"
    return parname


def ambient_labels(pname):
    if pname == "air_quality_index":
        return "Ambient air quality"
    elif pname == "noise_pollution":
        return "Ambient sound level"
    elif pname == "quality_of_surface_water_bodies":
        return "Ambient surface water quality"
    return pname


# method to save city to database
def save_city(request):
    city = str(request.POST.get('city'))
    cur = connection.cursor()
    cur.execute("INSERT INTO cities (name) values('"+city+"')")
    cur.close()
    return redirect('/solid')



def solid_result(request):
    #get the last city inserted
```

```
    con = connection.cursor()

    con.execute("select name from cities order by(id) desc limit 1")

    rs = con.fetchone()

    city = rs[0]


    # getting data of solid waste management

    Degree_of_Segregation = float(request.POST.get('Degree_of_Segregation'))

    Efficiency_in_collection_of_MSW =
float(request.POST.get('Efficiency_in_collection_of_MSW'))

    Extent_of_Solid_Waste_Recovered =
float(request.POST.get('Extent_of_Solid_Waste_Recovered'))

    Degree_of_Scientific_Disposal_of_MSW=
float(request.POST.get('Degree_of_Scientific_Disposal_of_MSW'))

    SWMP= float(request.POST.get('SWMP'))

    C_and_D= float(request.POST.get('C_and_D'))

    Extent_of_cost_recovery_in_SWM=
float(request.POST.get('Extent_of_cost_recovery_in_SWM'))


    # saving data recived from solid form to swm_Factor table in mysql

    cur = connection.cursor()

    cur.execute("INSERT INTO SWM_Factor
VALUES(NULL,'"+city+"',"+str(Degree_of_Segregation)+","+str(Efficiency_in_collecti
on_of_MSW)+","+str(Extent_of_Solid_Waste_Recovered)+","+str(Degree_of_Scientific
_Disposal_of_MSW)+","+str(SWMP)+","+str(C_and_D)+","+str(Extent_of_cost_recove
ry_in_SWM)+")")

    cur.close()


    c = connection.cursor()

    con = connection.cursor()

    con.execute("SELECT * FROM SWM_Factor WHERE city='"+city+"' order by(id)
desc")

    c.execute("select lower(parameter), weight from env_index where type
='%s'"%('Solid_waste_management'))
```

```
    row=c.fetchall()
    rs = con.fetchone()

    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))
    test =sorted(t.iteritems(), key = lambda x : x[1])
    dict_weights = convert_to_dict(row)
    swm_table_top = get_top_three(test,dict_weights)
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    swm_score = sum(final)*100
    swm_range_value = index_range(swm_score)

    return render(request,'solid_result.html',{'city':city,'name':'Solid Waste Management
Index','value':swm_score,'range':swm_range_value})



def water_result(request):
    #get the last city inserted
    con = connection.cursor()
    con.execute("select name from cities order by(id) desc limit 1")
    rs = con.fetchone()
    city = rs[0]
```

```
    Adequacy_of_water_supply = float(request.POST.get('Adequacy_of_water_supply'))

    Smart_meters_and_management =
float(request.POST.get('Smart_meters_and_management'))

    Leakage_identification = float(request.POST.get('Leakage_identification'))

    Continuity_of_water_supplied_in_hrs_per_day=
float(request.POST.get('Continuity_of_water_supplied_in_hrs_per_day'))

    Water_quality= float(request.POST.get('Water_quality'))

    Water_Recharge = request.POST.get('Water_Recharge')

    Extent_of_cost_recovery_in_WSM=
float(request.POST.get('Extent_of_cost_recovery_in_WSM'))



    # saving data from water form to WS_Factor table

    cur = connection.cursor()

    cur.execute("INSERT INTO WS_Factor
VALUES(NULL,'"+city+"',"+str(Adequacy_of_water_supply)+","+str(Smart_meters_an
d_management)+","+str(Leakage_identification)+","+str(Continuity_of_water_supplied_i
n_hrs_per_day)+","+str(Water_quality)+","+str(Water_Recharge)+","+str(Extent_of_cost
_recovery_in_WSM)+")")

    cur.close()



    c = connection.cursor()

    con = connection.cursor()

    con.execute("SELECT * FROM WS_Factor WHERE city='"+city+"' order by(id)
desc")

    c.execute("select parameter, weight from env_index where type
='%s'"%('Water_Supply'))

    row=c.fetchall()

    rs = con.fetchone()



    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))

    test =sorted(t.iteritems(), key = lambda x : x[1])

    dict_weights = convert_to_dict(row)
```

```
    ws_table_top = get_top_three(test,dict_weights)
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    ws_score = sum(final)*100
    ws_range_value = index_range(ws_score)
    return render(request,'water_result.html',{'city':city,'name':'Water Supply
Index','value':ws_score,'range':ws_range_value})



def sewage_result(request):
    #get the last city inserted
    con = connection.cursor()
    con.execute("select name from cities order by(id) desc limit 1")
    rs = con.fetchone()
    city = rs[0]



    Coverage_of_toilets=float(request.POST.get('Coverage_of_toilets'))

Collection_efficiency_of_Sewage_network=float(request.POST.get('Collection_efficienc
y_of_Sewage_network'))
```

```
Adequacy_of_sewage_treatment=float(request.POST.get('Adequacy_of_sewage_treatme
nt'))

    Quality_of_treated_sewage=float(request.POST.get('Quality_of_treated_sewage'))

    Wastewater_recycling=float(request.POST.get('Wastewater_recycling'))


Extent_of_cost_recovery_in_SSS=float(request.POST.get('Extent_of_cost_recovery_in_
SSS'))


Coverage_of_Storm_Water_Drainage=float(request.POST.get('Coverage_of_Storm_Wat
er_Drainage'))



    # code for saving data recived from sewage form to SS_Factor to database

    cur = connection.cursor()

    cur.execute("INSERT INTO SS_Factor
VALUES(NULL,'"+city+"',"+str(Coverage_of_toilets)+","+str(Collection_efficiency_of_
Sewage_network)+","+str(Adequacy_of_sewage_treatment)+","+str(Quality_of_treated_
sewage)+","+str(Wastewater_recycling)+","+str(Extent_of_cost_recovery_in_SSS)+","+s
tr(Coverage_of_Storm_Water_Drainage)+")")

    cur.close()



    c = connection.cursor()

    con = connection.cursor()

    con.execute("SELECT * FROM SS_Factor WHERE city='"+city+"' order by(id)
desc")

    c.execute("select parameter, weight from env_index where type
='%s'"%('Sewerage_and_sanitation'))

    row=c.fetchall()

    rs = con.fetchone()



    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))

    test =sorted(t.iteritems(), key = lambda x : x[1])

    dict_weights = convert_to_dict(row)

    ss_table_top = get_top_three(test,dict_weights)
```

```
    c.close()
    con.close()
    fac = []
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    ss_score = sum(final)*100
    ss_range_value = index_range(ss_score)
    return render(request,'sewerage_result.html',{'city':city,'name':'Sewerage Sanitation and
Storm Water Drainage Index','value':ss_score,'range':ss_range_value})



def enviroment_result(request):
    #get the last city inserted
    con = connection.cursor()
    con.execute("select name from cities order by(id) desc limit 1")
    rs = con.fetchone()
    city = rs[0]


    Ambient_air_quality=float(request.POST.get('Ambient_air_quality'))
    Ambient_sound_level=float(request.POST.get('Ambient_sound_level'))

Ambient_surface_water_quality=float(request.POST.get('Ambient_surface_water_quality
'))


    # code to dave data to AE_Facotr database
    cur = connection.cursor()
```

```
    cur.execute("INSERT INTO AE_Factor
VALUES(NULL,'"+city+"',"+str(Ambient_air_quality)+","+str(Ambient_sound_level)+"
,"+str(Ambient_surface_water_quality)+")")

    cur.close()



    c = connection.cursor()

    con = connection.cursor()

    con.execute("SELECT * FROM AE_Factor WHERE city='"+city+"' order by(id)
desc")

    c.execute("select parameter, weight from env_index where type
='%s'"%('Ambient_Environment'))

    row=c.fetchall()

    rs = con.fetchone()




    c = connection.cursor()

    con = connection.cursor()

    con.execute("SELECT * FROM AE_Factor WHERE city='"+city+"' order by(id)
desc")

    c.execute("select parameter, weight from env_index where type
='%s'"%('Ambient_Environment'))

    row=c.fetchall()

    rs = con.fetchone()


    t= dict(zip(map(lambda x:x[0].lower(), con.description[2:]), rs[2:]))

    test =sorted(t.iteritems(), key = lambda x : x[1])

    dict_weights = convert_to_dict(row)

    ae_table_top = get_top_three(test,dict_weights)


    c.close()

    con.close()

    fac = []
```

```
    for fval in row:
        fac.append(fval[1])
    final=[]
    m = 2
    for x in fac:
        final.append(float(rs[m])*float(x))
        m+=1
    ae_score = sum(final)*100
    ae_range_value = index_range(ae_score)
    return render(request,'enviroment_result.html',{'city':city,'name':'Ambient Environment Conditions Index','value':ae_score,'range':ae_range_value})
```

Solid Waste Management

```
{% include 'head.html'%}


<center>
    <!--<h1 style="text-align:center;">{{city}}</h1>-->
    <h3>Solid Waste Management Index</h3>
<table style="width:1000px;">
<tr>
<td style="width:400px;">
<form action="/solid/result/" method="POST">{% csrf_token %}
<input type="hidden" name="city" value="{{city}}"><br><br>



<!--{% for i in l  %}-->
<!--{{ i.0 }} ({{i.1}})<br>-->
    <!--<input type="text" name={{ i.0 }} required><br><br>-->


<!--{% endfor %}-->
```

```
<style>
    body {
    background-image:
    url("http://smartcitydss.pythonanywhere.com/static/img/3dgoGT.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 100% 100%;
    }
    </style>
<table>
   <tr>
      <td><label>Amount of MSW collected (TPD)</label></td>
      <td><label>Total amount of MSW generated (TPD)</label></td>
      <td><label> Efficiency in collection of MSW(EC) (0-1)</label></td>

   </tr>
   <tr>
      <td><input type="text" id="total_waste_collected" name="total_waste_collected"
required=""></td>
      <td><input type="text" id="waste_generated" name="waste_generated"
required=""></td>
      <td><input type="text" id="collresult"
name="Efficiency_in_collection_of_MSW"></td>
   </tr>
    <tr>
      <td><label>Amount of waste segregated (TPD)</label></td>
      <td><label>Total solid waste collected (TPD)</label></td>
      <td><label> Degree of Segregation(DS) (0-1)</label></td>

   </tr>
```

```
<tr>
<td><input type="text" id="waste_segregated" name="waste_segregated"
required=""></td>
<td><input type="text" id="waste_collected" name="waste_collected"
required=""></td>
<td><input type="text" id="segresult" name="Degree_of_Segregation" ></td>
</tr>

<tr>
<td><label>Amount of Waste Recycled(TPD)</label></td>
<td><label>Amount of MSW collected (TPD)</label></td>
<td><label>Extent of Solid Waste Recovered(SWR) (0-1)</label></td>

</tr>
<tr>
<td><input type="text" id="waste_recycled" name="waste_recycled"
required=""></td>
<td><input type="text" id="collected_waste" name="collected_waste"
required=""></td>
<td><input type="text" id="recresult"
name="Extent_of_Solid_Waste_Recovered"required=""</td>
</tr>
<tr>
<td><label>Amount of MSW disposed in sanitary landfills (TPD)</label></td>
<td><label>Total MSW disposed (TPD)</label></td>
<td><label> Degree of Scientific Disposal of MSW(SD) (0-1) </label></td>

</tr>
<tr>
<td><input type="text" id="Waste_disposed_in sanitary_landfill"
name="Waste_disposed_in sanitary_landfill" required=""></td>
```

```
      <td><input type="text" id="waste_disposed" name="waste_disposed"
required=""></td>

      <td><input type="text" id="sanresult"
name="Degree_of_Scientific_Disposal_of_MSW"required=""</td>

   </tr>

   <tr>

      <td colspan="3">

         Availability of separate system for recycling Construction & Demolition
waste(RCD)(0/1)<br> <input type = "radio" name="C_and_D" value="1"> yes  or <input
type="radio" name="C_and_D" value="0"> no

      </td>

   </tr>

    <tr>

      <td><label>Total revenues earned from MSWM (Lakh)</label></td>

      <td><label>Total expenses on MSWM (Lakh)</label></td>

      <td><label> Extent of cost recovery in SWM(CRSWM) (0-1)</label></td>


   </tr>

   <tr>

      <td><input type="text" id="Total_revenue" name="Total_revenue"
required=""></td>

      <td><input type="text" id="Total_expense" name="Total_expense"
required=""></td>

      <td><input type="text" id="cosresult"
name="Extent_of_cost_recovery_in_SWM"required=""</td>

   </tr>

   <tr>

      <td colspan='3'>

         SWM programs carried in the city during 3 years(SWMP) (0/1)<br> <input type
= "radio" name="SWMP" value="1"> yes or <input type="radio" name="SWMP"
value="0"> no<br>

      </td>

   </tr>
```

```
</table>
Type<br>
<input type="text" name=type value={{type}} readonly required><br><br>
   <input type="Submit" value="Submit">
</form>

</td>
<td style="width:650px;">
   Data is converted to a scale range of 0 to 1<br>
   Weights for each indicator is given by expert survey<br>
   Benchmarking and weights of indicators are shown in the table

<table border="1px" style="600px;border:1px solid black">

<tr>
<th>Indicators</th>
<th>Weights</th>
<th>Excellent</th>
<th>Good</th>
<th>Average</th>
<th>Poor</th>

</tr>
<tr>
<td>Efficiency in collection of MSW (EC)</td>
<td>0.155</td>
<td>>0.90</td>
```

<td>0.70-0.90</td>

<td>0.50-0.70</td>

<td>0-0.50</td>

</tr>

<tr>

<td>Degree of Segregation (DS)</td>

<td>0.171</td>

<td>>0.50</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

<td>0</td>

</tr>

<tr>

<td>Extent of solid waste recovered (SWR)</td>

<td>0.163</td>

<td>>0.50</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

<td>0</td>

</tr>

<tr>

<td> Degree of scientific disposal of MSW (SD)</td>

<td>0.165</td>

<td>>0.50</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

<td>0</td>

</tr>

<tr>

<td>Availability of separate system for recycling C & D waste (RCD)</td>

<td>0.133</td>

<td>1</td>

<td> </td>

<td> </td>

<td>0</td>

</tr>

<tr>

<td>Extent of cost recovery in Solid Waste Management (CRSWM)</td>

<td>0.13</td>

<td>>0.60</td>

<td>0.30-0.60</td>

<td>0-0.30</td>

<td>0</td>

</tr>

<td>SWM programs carried in the city during 3 years (SWMP)</td>

<td>0.083</td>

<td>1</td>

<td> </td>

<td> </td>

<td>0</td>

</tr>

</table>

View help

```
<center><a
href="http://mohua.gov.in/upload/uploadfiles/files/SLB%20National%20Data%20Book_
0.pdf">View Data Source</a></center>

</td>

</tr>


</table>

</center>


<script>

 function calculate(first,second,result){

       var value1 = document.getElementById(first).value;

       var value2 = document.getElementById(second).value;

       document.getElementById(result).value = value1/value2;

   }

 var waste =  document.getElementById('waste_collected');

 waste.addEventListener("focusout",function(){

     console.log('hello1');

     calculate('waste_segregated','waste_collected','segresult');

 });

document.getElementById('waste_generated').addEventListener("focusout",function(){

    console.log('hello2');

     calculate('total_waste_collected','waste_generated','collresult');


 });

 document.getElementById('collected_waste').addEventListener("focusout",function(){

    console.log('hello3');

     calculate('waste_recycled','collected_waste','recresult');


 });
```

```
 document.getElementById('waste_disposed').addEventListener("focusout",function(){
    console.log('hello4');
     calculate('Waste_disposed_in sanitary_landfill','waste_disposed','sanresult');
 });
document.getElementById('Total_expense').addEventListener("focusout",function(){
    console.log('hello5');
     calculate('Total_revenue','Total_expense','cosresult');
 });



</script>
```

Water Supply Management

```
{% include 'head.html'%}
<center>
<h3>Water Supply Management Index</h3>

<table style="width:1000px;">
    <tr>
    <td style="width:400px;">
<form action="/water/result/" method="POST">{% csrf_token %}
<input type="hidden" name="city" value="{{city}}"><br><br>




<!--{% for i in l  %}-->
<!--{{ i.0 }} ({{i.1}})<br>-->
    <!--<input type="text" name={{ i.0 }} required><br><br>-->

<!--{% endfor %}-->
```

```
<style>
    body {
    background-image:
    url("http://smartcitydss.pythonanywhere.com/static/img/3dgoGT.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 100% 100%;
    }
    </style>



<table>
   <tr>

      <td><label>Water Supplied (lpcd)</label></td>
      <td><label>Adequacy of water supply(AW) (0-1)</label></td>

   </tr>
   <tr>
      <td><input type="text" id="Water_Supply" name=" Water_Supply "
required=""></td>
      <td><input type="text" id="Per_Capita_Supply"
name="Adequacy_of_water_supply" required=""></td>

   </tr>

   <tr>
      <td><label> Number of metered connections</label></td>
      <td><label>Total water supply connections</label></td>
      <td><label>Smart meters and management(SMM) (0-1)</label></td>
```

```
    </tr>

    <tr>

        <td><input type="text" id="Total_number_of_ metered_connections"
name="Total_number_of_ metered_connections " required=""></td>

        <td><input type="text" id="total_connections" name="total_connections"
required=""></td>

        <td><input type="text" id="cmresult"
name="Smart_meters_and_management"required=""></td>


    </tr>

      <tr>

        <td><label>Volume of productive water (MLD)</label></td>

        <td><label>Total volume of water supply (MLD)</label></td>

        <td><label>Leakage identification(LI) (0-1)</label></td>


    </tr>

    <tr>

        <td><input type="text" id="water_sold" name="water_sold" required=""></td>

        <td><input type="text" id="water_in_distribution_system"
name="water_in_distribution_system" required=""></td>

        <td><input type="text" id="nrwresult"
name="Leakage_identification"required=""></td>

    </tr>

     <tr>


        <td><lable> Hours of water supplied(hrs)</lable></td>

        <td><label>Continuity of water supplied in hrs per day(CW) (0-1)</label></td>


    </tr>

    <tr>
```

```
        <td><input type="text" id="Number_of_hrs_supplied" name="
Number_of_hrs_supplied " required=""></td>

        <td><input type="text" id="Continuity"
name="Continuity_of_water_supplied_in_hrs_per_day" required=""></td>


    </tr>




    <tr>

        <td><label>Number of samples meeting or exceeding specified potable water
standards</label></td>

        <td><label>Total number of samples tested for water quality</label></td>

        <td><label>Water quality(WQ) (0-1)</label></td>


    </tr>

    <tr>

        <td><input type="text" id="samples_meeting_standards"
name="samples_meeting_standards" required=""></td>

        <td><input type="text" id="samples_tested" name="samples_tested"
required=""></td>

        <td><input type="text" id="qualresult" name="Water_quality"required=""></td>

    </tr>



     <tr>

        <td colspan='3'>

            Exploitation of underground water(EUGW) (0/1)<br> <input type="radio"
name="Water_Recharge" value="1"> yes or <input type="radio"
name="Water_Recharge" value="0"> no<br>

        </td>

    </tr>

    <tr>

        <td><label>Total revenue (Lakh)</label></td>
```

```
        <td><label>Total expense (Lakh)</label></td>
        <td><label>Extent of cost recovery in WSM(CRWS) (0-1)</label></td>

    </tr>
    <tr>
        <td><input type="text" id="Total_revenue" name="Total_revenue"
required=""></td>
        <td><input type="text" id="Total_expense" name="Total_expense"
required=""></td>
        <td><input type="text" id="cosrresult"
name="Extent_of_cost_recovery_in_WSM"required=""></td>
    </tr>
    </table>
Type<br>
<input type="text" name=type value={{type}} readonly required><br><br>
    <input type="Submit" value="Submit">

</form>
</td>
<td style="width:650px;">
    Data is converted to a scale range of 0 to 1<br>
    Weights for each indicator is given by expert survey<br>
    Benchmarking and weights of indicators are shown in the table

    <table border="1px" style="600px;border:1px solid black">
<tr>
<th>Indicators</th>
<th>Weights</th>
<th>Excellent</th>
<th>Good</th>
```

<th>Average</th>

<th>Poor</th>

</tr>

<tr>

<td>Adequacy of water supply(AW)</td>

<td>0.151</td>

<td>>0.75</td>

<td>0.75-0.50</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

</tr>

<tr>

<td>Smart meters and management(SMM)</td>

<td>0.145</td>

<td>>0.75</td>

<td>0.50-0.75</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

</tr>

</tr>

<td>Leakage identification(LI)</td>

<td>0.138</td>

<td>>0.80</td>

<td>0.70-0.80</td>

<td>0.70-0.50</td>

<td>< 0.50</td>

</tr>

<tr>

<td>Continuity of water supplied in hrs per day(CW)</td>

<td>0.127</td>

<td>>0.80</td>

<td>0.80-0.40</td>

<td>0.20-0.40</td>

<td>0-0.20</td>

</tr>

<tr>

<td>Water quality(WQ)</td>

<td>0.167</td>

<td>>0.90</td>

<td>0.70-0.90</td>

<td>0.40-0.70</td>

<td>0-0.40</td>

</tr>

<tr>

<td>Exploitation of underground water(EUGW)</td>

<td>0.163</td>

<td>1</td>

<td> </td>

<td> </td>

<td>0</td>

</tr>

<tr>

<td>Extent of cost recovery in WSM(CRWSM)</td>

<td>0.109</td>

<td>>0.75</td>

<td>0.50-0.75</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

```
</tr>

</table>

<center><a href="http://smartcitydss.pythonanywhere.com/static/help/help-water.pdf">View help</a></center>


<center><a href="http://mohua.gov.in/upload/uploadfiles/files/SLB%20National%20Data%20Book_0.pdf">View data source</a></center>


<center><a href="http://cgwb.gov.in/GW-data-access.html">View groundwater data source</a></center>

</td>

</tr>

</table>

</center>

<script>

 function calculate(first,second,result){

       var value1 = parseFloat(document.getElementById(first).value);

       var value2 = parseFloat(document.getElementById(second).value);

       document.getElementById(result).value = value1/value2;

   }

//  code for water supply

 document.getElementById('Water_Supply').addEventListener("focusout",function(){

     var result = 0.0;

     var water =  parseFloat(document.getElementById('Water_Supply').value);

     if(water>=135){

        result = 1.0;

     }else{

        result = water/135;

     }

     console.log(result);
```

```
    document.getElementById('Per_Capita_Supply').value = result;
});
//
//  code for Number of hrs supplied

document.getElementById('Number_of_hrs_supplied').addEventListener("focusout",func
tion(){
    var result = 0.0;
    var hrs=  parseFloat(document.getElementById('Number_of_hrs_supplied').value);
    if(hrs>=24){
        result = 1.0;
    }else{
        result = hrs/24;
    }
    console.log(result);
    document.getElementById('Continuity').value = result;
});

//
document.getElementById('total_connections').addEventListener("focusout",function(){
    calculate('Total_number_of_ metered_connections','total_connections','cmresult');
});


document.getElementById('water_in_distribution_system').addEventListener("focusout",
function(){
    calculate('water_sold','water_in_distribution_system','nrwresult');

});
```

```
 document.getElementById('samples_tested').addEventListener("focusout",function(){
    calculate('samples_meeting_standards','samples_tested','qualresult');

 });
document.getElementById('Total_expense').addEventListener("focusout",function(){
    calculate('Total_revenue','Total_expense','cosrresult');

 });
</script>
```

Sewerage, Sanitation and Stormwater drainage

```
{% include 'head.html'%}
<center>
<h3>Sewerage Sanitation and Storm water management Index</h3>
<table style="width:1000px;">
   <tr>
      <td style="width:400px;">
<form action="/sewage/result/" method="POST">{% csrf_token %}
<input type="hidden" name="city" value="{{city}}"><br><br>



<!--{% for i in l  %}-->
<!--{{ i.0 }} ({{i.1}})<br>-->
   <!--<input type="text" name={{ i.0 }} required><br><br>-->

<!--{% endfor %}-->
<style>
    body {
    background-image:
```

```
    url("http://smartcitydss.pythonanywhere.com/static/img/3dgoGT.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 100% 100%;
    }
    </style>


<table>
   <tr>
      <td><label>Volume of wastewater collected(MLD) </label></td>
      <td><label>Volume of total wastewater generated per day(MLD) </label></td>
      <td><label>Collection efficiency of Sewage network(CE)(0-1)</label></td>


   </tr>
   <tr>
      <td><input type="text" id="Total_waste_water_collected"
name="Total_waste_water_collected" required=""></td>
      <td><input type="text" id="Total_waste_water_generated"
name="Total_waste_water_generated" required=""></td>
      <td><input type="text" id="effresult"
name="Collection_efficiency_of_Sewage_network"></td>
   </tr>
   <tr>
      <td><label>Volume of  wastewater after Secondary treatment(MLD)</label></td>
      <td><label> Waste water generated (MLD)</label></td>
      <td><label> Adequacy of sewage treatment(AS) (0-1)</label></td>


   </tr>
   <tr>
      <td><input type="text" id="Secondary_treatment_of_waste_water"
name="Secondary_treatment_of_waste_water" required=""></td>
```

```
        <td><input type="text" id="Waste_water_generated"
name="Waste_water_generated " required=""></td>

        <td><input type="text" id="adresult"
name="Adequacy_of_sewage_treatment"></td>

    </tr>

     <tr>

        <td><label> Number of Treated wastewater samples which abide specified standards
</label></td>

        <td><label> Total number of treated wastewater samples</label></td>

        <td><label> Quality of treated sewage(QTS) (0-1)</label></td>



    </tr>

    <tr>

        <td><input type="text" id="Treated_waste_water_which_abide_standards"
name="Treated_waste_water_which_abide_standards" required=""></td>

        <td><input type="text" id="Total_waste_water_treated"
name="Total_waste_water_treated" required=""></td>

        <td><input type="text" id="quaresult" name="Quality_of_treated_sewage"></td>

    </tr>

    <tr>

        <td><label> Quantum of wastewater recycled(MLD)</label></td>

        <td><label> Total wastewater received at treatment plants(MLD)</label></td>

        <td><label>Wastewater recycling(WWR) (0-1)</label></td>



    </tr>

     <tr>

        <td><input type="text" id="Quantum_of_waste_water_recycled" name="
Quantum_of_waste_water_recycled " required=""></td>

        <td><input type="text" id="Total_waste_water_received_at_treatment_plants"
name="Total_waste_water_received_at_treatment_plants" required=""></td>

        <td><input type="text" id="rrresult" name="Wastewater_recycling"></td>

    </tr>
```

```
<tr>
<td><label>Total revenue(Lakh)</label></td>
<td><label> Total expense(Lakh)</label></td>
<td><label>Extent of cost recovery in SSS(CRSSS) (0-1)</label></td>
</tr>
<tr>
<td><input type="text" id="Total_revenue" name="Total_revenue"
required=""></td>
<td><input type="text" id="Total_expense" name=" Total_expense"
required=""></td>
<td><input type="text" id="crresult"
name="Extent_of_cost_recovery_in_SSS"></td>
</tr>

<tr>
<td><label>Total number of properties with access to toilets</label></td>
<td><label> Total number of properties </label></td>
<td><label> Coverage of toilets(CT) (0-1)</label></td>

</tr>
<tr>
<td><input type="text" id="Total_number_of_properties_with_access_to_toilets"
name="Total_number_of_properties_with_access_to_toilets" required=""></td>
<td><input type="text" id="Total_number_of_properties" name="
Total_number_of_properties" required=""></td>
<td><input type="text" id="tolresult" name="Coverage_of_toilets" ></td>
</tr>
 <tr>
<td><label>Total length of covered drains(Km)</label></td>
<td><label> Total length of road network(Km)</label></td>
<td><label>Coverage of Storm Water Drainage(CSWD) (0-1)</label></td>
```

```
    </tr>

    <tr>

        <td><input type="text" id="Total_length_of_covered_drains"
name="Total_length_of_covered_drains" required=""></td>

        <td><input type="text" id="Total_length_of_road_network"
name="Total_length_of_road_network" required=""></td>

        <td><input type="text" id="csdesult"
name="Coverage_of_Storm_Water_Drainage"></td>

    </tr>

</table>

Type<br>

<input type="text" name=type value={{type}} readonly required><br><br>

    <input type="Submit" value="Submit">

</form>


</td>

<td style="width:650px;">

    Data is converted to a scale range of 0 to 1<br>

    Weights for each indicator is given by expert survey<br>

    Benchmarking and weights of indicators are shown in the table



    <table border="1px" style="600px;border:1px solid black">

<tr>

<th>Indicators</th>

<th>Weights</th>

<th>Excellent</th>

<th>Good</th>

<th>Average</th>

<th>Poor</th>
```

</tr>

<tr>

<td>Collection efficiency of Sewage network(CE)</td>

<td>0.156</td>

<td>>0.70</td>

<td>0.40-0.70</td>

<td>0-0.40</td>

<td>0</td>

</tr>

<tr>

<td>Adequacy of sewage treatment(AS)</td>

<td>0.149</td>

<td>>0.70</td>

<td>0.40-0.70</td>

<td>0-0.40</td>

<td>0</td>

</tr>

<tr>

<td>Quality of treated sewage(QT)</td>

<td>0.152</td>

<td>>0.75</td>

<td>0.50-0.75</td>

<td>0.25-0.50</td>

<td>0-0.25</td>

</tr>

<tr>

<td>Wastewater recycling(WWR)</td>

<td>0.148</td>

<td>>0.20</td>

<td>0.10-0.20</td>

<td>0-0.10</td>

<td>0</td>

</tr>

<tr>

<td>Extent of cost recovery in SSS(CRSSS)</td>

<td>0.101</td>

<td>>0.60</td>

<td>0.30-0.60</td>

<td>0-0.30</td>

<td>0</td>

</tr>

<tr>

<td>Coverage of toilets(CT)</td>

<td>0.160</td>

<td>>0.90</td>

<td>0.70-0.90</td>

<td>0.35-0.70</td>

<td>0-0.35</td>

</tr>

<tr>

<td>Coverage of Storm Water Drainage(CSWD)</td>

<td>0.134</td>

<td>>1</td>

<td>0.60-1</td>

<td>0.60-0.30</td>

<td>0-0.30</td>

</tr>

```
</table>

   <center><a href="http://smartcitydss.pythonanywhere.com/static/help/help-
sewrage.pdf">View help</a></center>




<center><a
href="http://mohua.gov.in/upload/uploadfiles/files/SLB%20National%20Data%20Book_
0.pdf">View data source</a></center>

</td>

</tr>

</table>

<script>

function calculate(first,second,result){

      var value1 = document.getElementById(first).value;

      var value2 = document.getElementById(second).value;

      document.getElementById(result).value = value1/value2;

  }




document.getElementById('Total_number_of_properties').addEventListener("focusout",f
unction(){

    calculate ('Total_number_of_properties_with_access_to_toilets',
'Total_number_of_properties','tolresult');

});



document.getElementById('Total_waste_water_generated').addEventListener("focusout",
function(){

    calculate('Total_waste_water_collected','Total_waste_water_generated','effresult');

 });
```

```
document.getElementById('Waste_water_generated').addEventListener("focusout",functi
on(){

  calculate('Secondary_treatment_of_waste_water','Waste_water_generated','adresult');

 });


document.getElementById('Total_waste_water_treated').addEventListener("focusout",fun
ction(){


calculate('Treated_waste_water_which_abide_standards','Total_waste_water_treated','qua
result');

});


document.getElementById('Total_waste_water_received_at_treatment_plants').addEvent
Listener("focusout",function(){


calculate('Quantum_of_waste_water_recycled','Total_waste_water_received_at_treatment
_plants','rrresult');

});


document.getElementById('Total_expense').addEventListener("focusout",function(){

  calculate('Total_revenue','Total_expense','crresult');

});

document.getElementById('Total_length_of_road_network').addEventListener("focusout
",function(){

   calculate('Total_length_of_covered_drains','Total_length_of_road_network','csdesult');

});

</script>
```

Ambient Environment Condition

```
{% include 'head.html'%}


<center>
```

```
    <!--<h1 style="text-align:center;">{{city}}</h1>-->
    <h3>Ambient Environment Condition Index for city : {{city}}</h3>
<style>
    body {
    background-image:
    url("http://smartcitydss.pythonanywhere.com/static/img/3dgoGT.jpg");
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 100% 100%;
    }
    </style>
<table border="1">
        <tr>
                <th>Indicators</th>
                <th>Score</th>
                <th>Rating</th>
        </tr>
   <tr>
      <td>{{name}}</td>
      <td>{{value}}</td>
      <td>{{range}}</td>
   </tr>
</table>
</center>
</table>
            <hr>
        </td>
        </tr><tr>
        <td style"width:500px;">
```

<h2 style="text-align:center;">Action Plan for Ambient Environment Condition</h2>

<table style"width:500px;" border="1px">

<tr>

<th>Indicators</th>

<th>Action Plan</th>

<th>Approximate cost analysis</th>

</tr>

<tr>

<td>Ambient Air Quality(AAQ)</td>

<td>Vehicular emissions and industrial emissions can be controlled by use of catalytic converter and installation of pollution abatement techniques or use of natural gas respectively. (Greenstone et.al, 2017)</td>

<td>NA</td></tr>

<tr>

<td>Ambient sound level(ASL)</td>

<td>Installation of noise barriers and sound absorptive materials.(Greenstone et.al, 2017) </td>

<td>NA</td>

</tr>

<tr>

<td rowspan="2">Ambient Surface Water Quality(ASW)</td>

<td>Rehabilitation Of Existing Water Sources(Surface Subsurface) (Amrut Lucknow, 2015)</td>

<td>60cr for increment from 73% to 95%</td>

<tr>

<td>Restoration Of Water Bodies (Lake,Ponds,Talabs) (Amrut Lucknow, 2015)</td>

<td>60cr for increment from 73% to 100% </td>

</tr>

</table>