Annotated Code Document

Project: Automation of Concrete Mix Design Model

Client: Nebraska Department of Transportation (NDOT)

Code	Explanation
import pandas as pd import numpy as np	These import common Python libraries needed to run the code
project_name = "Project #2: Automation of NDOT Concrete Mix Design" client = "Nebraska Department of Transportation (NDOT)"	Stores project and client info as variables. They're printed later so outputs are clearly tied to the correct project
print(f'This project is designed for: {client}')	Prints the client name using an f-string. Demonstrates formatted output in Python
my_number = 7.341212 print(f"{my_number:8.2f}")	Demonstrates numeric formatting. Rounds a number to two decimal places. Used later for clean output formatting
cubic_yard_ft3 = 27 unit_weight_water = 62.4	Defines fixed constants for the entire code. One cubic yard is 27 cubic feet and the unit weight of water is 62.4 lb per cubic feet. Since its used repeatedly throughout code defining it once will make things simpler
print(f"defined constant: 1 cubic yard = {cubic_yard_ft3} cubic feet") print(f"Unit weight of water = {unit_weight_water} lb per cubic feet")	Prints the defined constants to confirm the values

def calculate_water_weight_Q(cement_A, fly_ash_B, silica_fume_C, other_SCM_D, wc_ratio_E): total_cementious = cement_A + fly_ash_B + silica_fume_C + other_SCM_D water_weight_Q = total_cementious * wc_ratio_E return water_weight_Q	Calculates the total water weight for the mix. First it lists all the materials needed. Then it defines total_cementious as the sum of A, B, C, and D. then it multiples itself by E to achieve Q. Returning Q sends the calculated value back to the main program
w_weight_1 = calculate_water_weight_Q(600, 100, 30, 70, 0.42) print(f"Water weight = {w_weight_1} lb/yd^3")	An example used for the function using random numeric values and printing the calculated water weight
cement_A = 600 fly_ash_B = 100 silica_fume_C = 30 other_SCM_D = 70 wc_ratio_E = 0.42	Demonstrates using variables as inputs.
w_weight_2 = calculate_water_weight_Q(cement_A, fly_ash_B, silica_fume_C, other_SCM_D, wc_ratio_E) print(f"water weight = {w_weight_2} lb/yd^3")	Both methods give the same output when ran

def calculate_volume_cement_R(cement_A, sg_cement_J):
volume_R = cement_A / (sg_cement_J * unit_weight_water)
return volume_R
# Example input values for demonstration 	
cement_A = 600
sg_cement_J = 3.5
# Use them in the function 	
R = calculate_volume_cement_R(cement_A, sg_cement_J)
# Print 	
print(f"cement volume (R): {R:.3f} ft^3")
	Calculates all the different material volumes using the same formula of volume = weight divided by (specific gravity * unit weight of water)

def calculate_volume_fly_ash_S(fly_ash_B, sg_fly_ash_k): return fly_ash_B / (sg_fly_ash_k * unit_weight_water)	Structurally the same as the previous volume function

def calculate_volume_silica_fume_T(silica_fume_C, sg_silica_fume_L):
  return silica_fume_C / (sg_silica_fume_L * unit_weight_water)
 	
def calculate_volume_SCM_U(other_SCM_D, sg_other_SCM_M):
  return other_SCM_D / (sg_other_SCM_M * unit_weight_water)	

def calculate_air_volume_V(air_content_F):
  volume_V = (air_content_F /100) * cubic_yard_ft3
  return volume_V
def calculate_water_volume_W(weight_water_Q):
  volume_W = water_weight_Q / unit_weight_water
  return volume_W
	Calculates the air and water volumes.

def calculate_total_aggregate_volume_X(volume_R, volume_S, volume_T, volume_U, volume_V, volume_W):
  volume_X = (cubic_yard_ft3 - volume_R - volume_S - volume_T - volume_U - volume_V - volume_W)
  return volume_X
	Calculates the remaining volume available for aggregates. Subtracts all the cementitious, air, and water volumes from 27 cubic feet. Each function uses the aggregate percentage, specific gravity, and converts the volume to weight

def calculate_fine_aggregate_weight_Y(percent_fine_G, sg_fine_N, volume_X):
  weight_Y = unit_weight_water * (percent_fine_G / 100) * sg_fine_N * volume_X
  return weight_Y
 	
def calculate_coarse_aggregate_Z(percent_coarse_H, sg_coarse_O, volume_X):
  return unit_weight_water * (percent_coarse_H / 100) * sg_coarse_O * volume_X
 	
def calculate_other_aggregate_AA(percent_other_I, sg_other_P, volume_X):
  return unit_weight_water * (percent_other_I / 100) * sg_other_P * volume_X
	Structurally the same as the previous aggregate volume function

# Scenario 1 – Standard Pavement Mix 	
project_no = 101
concrete_class = "47B"
	User input example

cement_A = 600 fly_ash_B = 100 silica_fume_C = 30 other_SCM_D = 70 water_cement_ratio_E = 0.42 air_content_F = 6 percent_fine_G = 45 percent_coarse_H = 50 percent_other_I = 5 sg_cement_J = 3.15 sg_fly_ash_K = 2.30 sg_silica_fume_L = 2.20 sg_other_SCM_M = 2.60 sg_fine_N = 2.65 sg_coarse_O = 2.70 sg_other_P = 2.60	

# Start with a general explanation 	
print("\n----------")
print(" NDOT Concrete Mix Design – Weight Summary")
print(" (1 Cubic Yard of Concrete)")
print("----------")
	print() outputs text to the console. \n inserts a blank line before the first divider. The dashed lines separate sections

# Project and mix details 	
print(f"Project Number: {project_no}")
print(f"Class of Concrete: {concrete_class}")
print("----------")
	f”” is an f-string. {project_no} and {concrete_class} insert variable values. This identifies info for the mix.

# Cementitious material outputs 	
print(f"Cement (A): {cement_A:8.1f} lb")
print(f"Fly Ash (B): {fly_ash_B:8.1f} lb")
print(f"Silica Fume (C): {silica_fume_C:8.1f} lb")
print(f"Other SCM (D): {other_SCM_D:8.1f} lb")
print("----------")
	These print cementious material weights. The :8.1f means that the total width is 8 characters with 1 decimal place and a fixed point format. This keeps the numbers in columns. Lb means in pounds

# Aggregate outputs 	
print(f"Fine Aggregate (Y): {Y:8.0f} lb")
print(f"Coarse Aggregate (Z): {Z:8.0f} lb")
print(f"Other Aggregate (AA): {AA:8.0f} lb")
print("----------")
	Prints aggregate weights. :8.0f means its an 8 character width with 0 decimal places and rounded to the whole number

# Water output 	
print(f"Water (Q): {Q:8.0f} lb")
	Prints total batch water

# End it with a note 	
print("----------")
print("End of Mix Design Summary")
	Final divider line

2nd block is the same so no need to repeat. When running every piece of code and inputting, you will get a final chart that displays everything	