COLLEGE CODE : 1128

COLLEGE NAME : TJS ENGINEERING COLLEGE

DEPARTMENT : COMPUTER SCIENCE ENGINEERING

STUDENT NM – ID : aut112823CSE25

aut112823CSE26

aut112823CSE22

aut112823CSE36

aut112823CSE45

ROLL NO : 112823104025

112823104026

112823104022

112823104036

112823104045

COMPLETED THE PROJECT NAMED AS :

URBAN PLANNING AND DESIGN

SUBMITTED BY ,

NAME	MOBILE NO
KAMALI .P	9042716176
KANISHMA KAVIYA M.S	9585595725
JOSHNA .S	8754659024
LALASA .D	8807769779
MONIKA .M.G	9578794277

Phase 4: Performance of the Project

Title: Urban Planning and Design

Objective

The objective of this project is to simulate and analyze an urban environment using Python. The project aims to demonstrate key urban planning principles including zoning, green space distribution, and optimized land usage through a simplified 10x10 city grid model. 1. Simulation Design

Overview:

A simulated urban grid is constructed where each cell represents a city unit. Buildings (B), parks (P), and empty land (.) are distributed according to predefined rules that reflect real-world urban planning standards.

Implementation:

This simulation presents a clear representation of an urban zone with balanced distribution of residential and green areas. The model serves as a foundation for further development in traffic, population, and infrastructure planning.

Outcome:

The Python script uses only standard libraries to ensure compatibility across platforms. It randomizes placement of buildings and parks while respecting spacing rules.

Legend: B = Building, P = Park, . = Empty Land

Example Grid:

B B . P B B B B B B

B B B B B B B B B B

B B B B B B B B . B

B B B B B B B B B B

B B B B . B B B B B

B . B B B B B B B B

B B B B B B B P B B B

B B B B B B B B B

B B B B B B B B B B

B B B B B B B B B B

4. Outcomes of Phase 3

- Implemented a working urban simulation model.

- Integrated planning rules and spatial constraints in a visual format.

- Created opportunities for extension into population modeling, traffic simulation, and infrastructureplanning.

5. Future Scope (Phase 4)

Introduce zoning for commercial, residential, and industrial areas.
Add transportation routes and simulate traffic flow.

Enhance visuals with GUI using Tkinter or Pygame.

Integrate real-world data for smart city planning simulation.

Source code

```
import random

# Define city grid
rows, cols = 10, 10
city_grid = [['Empty' for _ in range(cols)] for _ in range(rows)]

# Function to place buildings
def place_buildings(n_buildings):
    count = 0
    while count < n_buildings:
        r, c = random.randint(0, rows - 1), random.randint(0, cols - 1)
        if city_grid[r][c] == 'Empty':
            city_grid[r][c] = 'B'  # B = Building
            count += 1

# Function to check if a park can be placed
def is_valid_park_location(r, c):
    for i in range(max(0, r - 1), min(rows, r + 2)):
        for j in range(max(0, c - 1), min(cols, c + 2)):
            if city_grid[i][j] == 'P':
                return False
    return True

# Function to place parks
def place_parks(n_parks):
    count = 0
    while count < n_parks:
```

```
        r, c = random.randint(0, rows - 1), random.randint(0, cols - 1)
        if city_grid[r][c] == 'Empty' and is_valid_park_location(r, c):
            city_grid[r][c] = 'P'  # P = Park
            count += 1

# Function to display grid
def display_city_grid():
    print("\nUrban City Grid Layout:")
    print("Legend: B=Building, P=Park, .=Empty\n")
    for row in city_grid:
        for cell in row:
            if cell == 'Empty':
                print('.', end=' ')
            else:
                print(cell, end=' ')
        print()

# Place buildings and parks
place_buildings(60)
place_parks(10)

# Display the city grid
display_city_grid()
```

Output:

Urban Planning and Design - Grid Simulation

Legend: B=Building, P=Park, .=Empty Land

B B B . B P B B B B

B B B B B B B B B B

B B B . B B B B B B

B B B B B B B B B B

B B B B B B . B B B

B . B B B B B B B B

B B B B B B B P B B

B B B B B B B B B B

B B B B B B B B B B

B B B B B B B B B B