**Task 3: Data Visualization – Generate 3D Surface Plots with Custom Color Maps and Shading**

**Objective:**

The objective of this task was to generate 3D surface plots using Python, with customized color maps and shading to enhance the visualization of mathematical functions. The plots were designed to represent various surface functions over a 2D grid, with different shading effects and color maps applied to each surface plot.

**Methodology:**

1. **Data Preparation:**
   * Generated a meshgrid using numpy's linspace() and meshgrid() functions to create a 2D grid of values for X and Y.
   * Three different mathematical functions were used to generate the corresponding Z values:
     1. **Function 1**: Z1 (sin(sqrt(X^2 + Y^2)))
     2. **Function 2**: Z2 (cos(X) \* sin(Y))
     3. **Function 3**: Z3 (exp(-0.1 \* (X^2 + Y^2)))
2. **Surface Plot Generation:**

* Used Matplotlib’s Axes3D for 3D plotting.
* Each surface plot was created using the plot\_surface() method with different color maps and shading options:
  + **Color Maps Used**:
    - 'viridis' for the first plot.
    - 'inferno' for the second plot.
    - 'plasma' for the third plot.
  + **Shading**: Enabled shading (shade=True) to give depth to the surfaces and make them visually more dynamic.

1. **Customization**:

* **Color Bar**: A color bar was added to each subplot to indicate the mapping between colors and Z-values.
* **Titles and Labels**: Each plot was labeled with a title, and axis labels were added for clarity.
* **Layout**: Used plt.tight\_layout() to ensure proper spacing between subplots and prevent overlap.

1. **Results:**

The generated 3D surface plots showed three different mathematical surfaces, each with a distinct color map and shading. The color maps provided different visual effects, and the shading enhanced the depth perception of the 3D surfaces.

