**Project Summary: Terrain Generation using Perlin Noise**

**Title:** Terrain Generation using Perlin Noise

**Group Members:**

* Hamna Mudassar (FA22-BCS-125)
* Sadia Usman (FA22-BCS-129)

**Serial Complexity of the Code:** O(N^2), where N is the grid size (250x250). Terrain generation involves iterating over each grid point and applying Perlin noise, which is computationally expensive without multithreading.

**Parallel Complexity of the Code:** O(N^2 / T), where T is the number of CPU threads used. Multithreading via Python’s threading module allows the terrain to be computed in parallel row-wise, significantly reducing wall-clock time.

**Work Distribution:**

* Hamna Mudassar: Implemented generation of all terrain types (Mountains, Hills, Plains, Islands) and developed the full GUI interface.
* Sadia Usman: Implemented generation of all the maps (Biome Map, Height Map, Temperature Map, Moisture Map, Topographic Map).

**External Language Used:** Python

**Reason for Not Using C++:** C++ implementation required Microsoft Build Tools, which could not be installed due to system restrictions. The backup plan was to use WSL (Ubuntu), but GUI display of terrains required Xlaunch, which also failed. Due to these limitations, the project was shifted to Python.

**Workflow of the App:**

1. **Terrain Generation:**
   * Different terrains are generated using the algorithm discussed in slides
   * Terrain is generated using Perlin noise with terrain-specific postprocessing.
   * Parallelization is achieved by implementing multithreading
2. **Maps Generation:**
   * **Biome Map:** Visualizes terrain zones like water, sand, grass, rock, and snow.
   * **Height Map:** Shows elevation as colors.
   * **Temperature Map:** Inversely related to height (higher = colder).
   * **Moisture Map:** Derived from height with a squared decay for realism.
   * **Topographic Map:** Shows contour lines for elevation differences.
3. **GUI:**
   * The GUI is implemented using tkinter from Python.
   * Layout includes buttons for terrain selection, map viewing, and user guide.
   * Thread usage and complexity info is displayed via a pop-up.

**4.Summary:**

Different types of terrains are generated in parallel along with different maps which explains different aspects of terrains