**Task: Clustering Basketball Shots**

**Objective:**

Break down basketball shooting sessions into meaningful clusters based on shot location and time proximity. Clusters will represent groups of shots taken from approximately the same court location within a reasonable time frame. The goal is to explore different clustering approaches, leveraging large datasets and visual validation.

**Input:**

1. **Shot Data**: Each shot will have metadata, including shot location (2D court coordinates) and shot time.
   * Data can be accessed via:
     + CSV files (initial focus).
     + API (for future integration).
2. **Sessions**: A session consists of multiple shots (can range from 10s to 100s).

**Challenges:**

1. **Clustering Shots**: Sessions need to be segmented into meaningful clusters. Clusters will have approximately 10-20 shots taken from a similar location and time frame. The time interval between shots won’t be uniform and will need to be determined through experimentation.
2. **Deciding the Number of Clusters**: Clustering is subjective, and different interpretations may exist for the same session. The clustering algorithms should aim for consistent and logical grouping but remain configurable for different scenarios.

**Approach:**

1. **Clustering Methods**:
   * Experiment with various clustering algorithms, such as K-means, DBSCAN, and hierarchical clustering, in 2D space (shot location) and over time (shot timing).
   * Test combinations of spatial and temporal clustering.
2. **Advantages**:
   * Leverage the large datasets available.
   * Visual validation will be done through static images, providing simplicity for early-phase research.

**Implementation Overview:**

* **Programming Language**: Python.
* **Configuration Files**:
  + **Code Configuration**: Specifies data source (CSV or API), relevant credentials, and paths.
  + **Clustering Configuration**: Parameters for clustering algorithms, minimum and maximum shots per cluster, time intervals, and chosen algorithm(s).
* **Modules**:
  + **Shot Getter**: Retrieves shot data from CSV or API (initial focus is CSV).
  + **Clusterizer**: Uses clustering algorithms to segment shots into clusters.
  + **Visualizer**: Displays clustering results in simple, static visual outputs (e.g., static court map with timestamps and shot locations).

**Data to be Provided:**

* Sample CSV files with real session data.
* Documentation on shot formats, visual display standards, and other relevant details as needed.