- API Documentation: Running Tracker
  - start_run
    - Returns:
    - Example:
  - end_run
    - Parameters:
    - Returns:
    - Example:
  - store_run
    - Parameters:
    - Example:
  - check_alerts
    - Parameters:
    - Example:
  - get_runs
    - Returns:
    - Example:
  - generate_table
    - Parameters:
    - Example:
    - Output:
  - generate_graph
    - Parameters:
    - Example:
    - Description:
  - Database Schema
  - Configuration
    - Global Variables:
    - Example:

API Documentation: Running Tracker

This API facilitates tracking, storing, and analyzing runs. It includes functions for managing sessions, storing data in a SQLite database, and visualizing results.

`start_run`

Starts a running session by recording the current time.

Returns:

- `start_run_time` *(float)*: The Unix timestamp of when the run started.

Example:

```
start_time = start_run()
print(f"Run started at: {start_time}")
```

`end_run`

Ends a running session, calculates metrics, and stores the run data in the database.

Parameters:

- `distance` *(float)*: Distance of the run in meters.

Returns:

- A tuple containing:
  - `time_spent` *(float)*: Duration of the run in minutes.
  - `average_speed` *(float)*: Average speed in kilometers per hour.

Example:

```
time_spent, avg_speed = end_run(5000)  # 5000 meters
```

```
print(f"Time: {time_spent:.2f} minutes, Speed: {avg_speed:.2f} km/h")
```

`store_run`

Stores the run data in the SQLite database.

Parameters:

- `distance` *(float)*: Distance in meters.
- `time` *(float)*: Time in minutes.
- `average_speed` *(float)*: Average speed in kilometers per hour.

Example:

```
store_run(5000, 25.0, 12.0)
print("Run data stored successfully.")
```

`check_alerts`

Checks if the average speed falls within a specified range, and prints alerts if necessary.

Parameters:

- `average_speed` *(float)*: Average speed in kilometers per hour.
- `min_speed` *(float, optional)*: Minimum speed threshold. Defaults to `MIN_SPEED`.
- `max_speed` *(float, optional)*: Maximum speed threshold. Defaults to `MAX_SPEED`.

Example:

```
check_alerts(10.5)  # Example speed within default limits
check_alerts(3.5, min_speed=5)  # Custom minimum speed threshold
```

---

get_runs

Retrieves all stored runs from the database, ordered by date in descending order.

Returns:

- A list of tuples, where each tuple contains:
  - `id` *(int)*: Run ID.
  - `date` *(str)*: Date and time of the run.
  - `distance` *(float)*: Distance in meters.
  - `time` *(float)*: Duration in minutes.
  - `average_speed` *(float)*: Average speed in kilometers per hour.

Example:

```
runs = get_runs()
for run in runs:
    print(run)
```

---

generate_table

Prints a table of all runs with customizable column labels.

Parameters:

- `label_coluns` *(list)*: A list of strings specifying column labels in order: `["ID", "Date", "Distance", "Time", "Average Speed"]`.

Example:

```
generate_table(["ID", "Date", "Distance (m)", "Time (min)", "Speed (km/h)"])
```

Output:

```
 Date: 2024-11-19 08:30:00, Distance (m): 5000, Time (min): 25.00, Speed (km/h):
12.00
 Date: 2024-11-18 07:45:00, Distance (m): 3000, Time (min): 20.00, Speed (km/h):
9.00
```

`generate_graph`

Generates a line graph showing the average speed over time with customizable labels and markers.

Parameters:

- `x_label` *(str)*: Label for the x-axis.
- `y_label` *(str)*: Label for the y-axis.
- `title` *(str)*: Title of the graph.
- `marker` *(str)*: Marker style for the graph, e.g., `"o"` for circular markers.

Example:

```
generate_graph("Date", "Average Speed (km/h)", "Speed Progress Over Time",
marker="o")
```

Description:

- Plots average speed (`y-axis`) against the date of each run (`x-axis`).
- Ensures proper formatting for readability.

Database Schema

The SQLite database (`runs.db`) contains a single table named `runs` with the following schema:

- `id` *(INTEGER PRIMARY KEY)*: Unique identifier for each run.
- `date` *(TEXT)*: Date and time of the run in the format `YYYY-MM-DD HH:MM:SS`.
- `distance` *(REAL)*: Distance of the run in meters.
- `time` *(REAL)*: Duration of the run in minutes.
- `average_speed` *(REAL)*: Average speed in kilometers per hour.

Configuration

Global Variables:

- `MIN_SPEED` *(float)*: Minimum acceptable speed for a run, in kilometers per hour. Default: `4`.
- `MAX_SPEED` *(float)*: Maximum acceptable speed for a run, in kilometers per hour. Default: `17`.

Example:

```
MIN_SPEED = 5  # Adjust minimum speed to 5 km/h
MAX_SPEED = 15  # Adjust maximum speed to 15 km/h
```
