Week 1 Progress Report

DONE Week 1 Progress Summary

1. Virtual Environment & Setup

- Created a clean, isolated Python `.venv` for the project.
- Installed and locked required libraries (`numpy`, `matplotlib`, `pytest` — ready for later phases).
- Added a clear **setup guide (PDF)** for team members to reproduce the same environment easily.

2. Dataset Generator (DataLoader)

- Implemented `DatasetGenerator` to produce test data sets ranging from **1 → 1,000,000 keys**.
- Supports multiple data patterns:
    - **Sequential** (best case)
    - **Uniform random** (average case)
    - **Mixed/clustered** (more realistic, uneven distribution)
- Used NumPy for reproducibility and speed.
- Confirmed data generation and sorting via a built-in tester.

3. Baseline B-Tree Implementation

- Built a **read-optimized B-Tree** with:
    - `build_from_sorted_array()` – bulk load from sorted keys
    - `search()` – logarithmic lookup using binary search
    - `get_memory_usage()` – rough per-node memory estimate

- Tested interactively to confirm correct lookups and structural integrity.
- Ready for later extension (insert/delete logic for dynamic tests).

4. **Benchmarking Framework (Phase 1)**
   - Developed a benchmark script to:
     - Measure **build time**, **average lookup time**, and **memory use**.
     - Compare performance for different **page sizes** (32, 64, 128, 256).
   - Implemented `Benchmark.run()` to generate test queries (half existing, half random) and report timing results.
   - Integrated it into a simple `main.py` driver for fully automated testing across datasets and sizes.

5. **Validation**
   - Verified both the DataLoader and B-Tree modules run correctly inside the virtual environment.
   - Confirmed imports and package structure (`src/__init__.py` etc.) so the project runs cleanly as a package.

Next Steps (Week 2 Preview)

- Extend the benchmarking system to **save results (CSV)** for plotting.
- Implement the **Linear Regression Index (learned index)**.
- Compare learned vs B-Tree on the same datasets.
- Begin collecting graphs for build time, lookup latency, and memory trade-offs