Mini Search Engine Project – Extended Roadmap (Advanced Phases)

These phases extend the original roadmap into an advanced, top-tier systems project suitable for interviews, final-year projects, and performance-oriented system design discussions.

Phase 11: Persistent Storage (File-Based Indexing)

- Serialize the inverted index to disk using binary files.
- On program startup, check if index file exists.
- Load index directly instead of reprocessing documents.
- Demonstrates data persistence and efficient startup design.

Phase 12: Multi-Threaded Indexing

- Use std::thread or std::async to index multiple files concurrently.
- Split document set across CPU cores.
- Measure indexing time before and after parallelism.
- Demonstrates systems programming and performance optimization.

Phase 13: Text Normalization (Stemming & Stop Words)

- Implement stemming (e.g., Porter Stemmer) to reduce words to root form.
- Ignore stop words like 'the', 'is', 'at' using a predefined set.
- Improves index quality and search relevance.
- Demonstrates NLP-aware preprocessing.

Phase 14: Advanced Ranking (TF-IDF)

- Implement Term Frequency–Inverse Document Frequency scoring.
- TF = (Word frequency in document / Total words in document).
- IDF = log(Total documents / Documents containing word).
- Rank results using TF-IDF score.
- Demonstrates applied mathematics and information retrieval theory.

Phase 15: Boolean & Proximity Search

- Support AND / OR boolean queries.
- Parse user queries into logical components.
- Return documents matching query logic.
- Optional: proximity-based search (words within N distance).
- Demonstrates query parsing and compiler-like thinking.

Phase 16: Performance & Scalability Analysis

- Analyze time complexity of indexing and querying.
- Test with large document collections.
- Visualize performance improvements (optional graphs).
- Demonstrates scalability awareness.

Phase 17: Final Documentation & Interview Readiness

- Document system architecture and design trade-offs.
- Explain why inverted index is superior to brute force.
- Prepare concise explanations for interviews.
- Position project as systems + algorithms focused.