Spark Project Readiness Preparation Plan

Phase 1: Foundations (1–2 weeks)

Objective: Build strong basics in Python, distributed computing, and Spark fundamentals.

- **Python Prep (Data Engineering Focus):**
    - Data structures (lists, dicts, sets, tuples).
    - File handling (CSV, JSON, Parquet).
    - Functions, OOP basics, error handling.
    - Libraries: `pandas`, `os`, `logging`.
- **Distributed Systems Intro:**
    - Why Spark vs. Hadoop?
    - MapReduce concept.
    - Spark vs. Pandas.
- **Spark Fundamentals:**
    - RDD vs DataFrame vs Dataset.
    - Lazy evaluation, DAG, Transformations & Actions.
    - Spark Shell, SparkContext, SparkSession.

Practice Programs:

```
# Word count with RDD
from pyspark import SparkContext
sc = SparkContext("local", "WordCount")
rdd = sc.textFile("data.txt")
word_counts = rdd.flatMap(lambda line: line.split()) \
                 .map(lambda word: (word, 1)) \
                 .reduceByKey(lambda a, b: a + b)
print(word_counts.collect())
```

Phase 2: Core Spark (2–3 weeks)

Objective: Ability to code transformations, actions, and data pipelines.

- **Spark SQL & DataFrames:**
    - Loading data from CSV, JSON, Parquet.
    - Select, filter, groupBy, agg.
    - Joins, Window functions, UDFs.

- **Spark Architecture:**
    - Driver, Executors, Cluster Manager.
    - Jobs, Stages, Tasks.
- **Coding Best Practices:**
    - Partitioning, caching, shuffling.
    - Narrow vs wide transformations.

Practice Programs:

- Load Sales CSV → clean nulls → calculate revenue by region.
- Join Customer & Orders datasets → get top 5 customers by spend.
- Implement custom UDF (e.g., categorize age groups).

Phase 3: Advanced Spark (2–3 weeks)

Objective: Efficient pipelines, optimization, debugging, troubleshooting.

- **Performance:**
    - Spark UI – reading jobs/stages.
    - Tungsten, Catalyst Optimizer.
    - Partition tuning, broadcast joins.
- **Advanced APIs:**
    - Window functions (ROW_NUMBER, RANK).
    - Explode, Structs, Nested JSON parsing.
- **Streaming:**
    - Structured Streaming basics.
    - Kafka integration.

Practice Programs:

- Write batch ETL: Read logs, parse JSON, aggregate metrics, write Parquet.
- Implement Spark Streaming job with Kafka.
- Optimize join with broadcast hint.

Phase 4: Deployment Readiness (2–3 weeks)

Objective: End-to-end pipeline, troubleshooting, project simulation.

- **Project Simulation:**
    - Ingest → Transform → Aggregate → Store in warehouse (e.g., PostgreSQL/S3).
    - Add checkpoints, logging, error handling.
- **Troubleshooting Skills:**

    - Debug Spark job failures (out of memory, skew, stage retries).
    - Monitor Spark UI metrics.
- **Team Readiness:**
    - Code reviews (efficiency, readability).
    - Unit testing with `pytest` + `chispa`.

Practice Project Idea:

- **Retail ETL Project:**
    - Source: sales transactions (CSV/JSON).
    - Tasks: clean, transform, aggregate, load.
    - Tech: PySpark + SparkSQL + S3/Postgres.
    - Deliverables: pipeline code, test cases, README, troubleshooting notes.

Materials & References

- **Books:**
    - *Learning Spark, 2nd Edition* (O’Reilly).
- **Docs:**
    - PySpark API Docs
- **GitHub Repos:**
    - awesome-spark
    - databricks spark-examples
- **Practice Data:**
    - Kaggle: Retail, E-commerce, MovieLens datasets.

Metrics for Evaluation

Can write RDD, DataFrame, SQL pipelines.
Understand Spark architecture (driver, executors).
Handle joins, windowing, aggregations.
Debug & optimize Spark jobs.
Build an end-to-end ETL pipeline.