🧩 Day 1 – Scala Foundations for Data Engineers

Topics:

- Overview: Scala in the Big Data Ecosystem
  (Why Scala for Spark, Hadoop, Hive, Kafka, Delta Lake, etc.)
- Setting up environment: JDK 11 + Scala 2.12 + SBT + IntelliJ + Spark 3.3.4
- Basic Syntax and REPL
- Variables, Data Types, and Operators
- Control Structures (if, match, for, while)
- Functions and Recursion
- Collections (List, Set, Map, Seq)
  → Functional transformations (`map`, `filter`, `reduce`)
- Tuples and Case Classes
- Pattern Matching
- Error Handling (`Option`, `Try`, `Either`)
- Hands-on:
    - Transforming and cleaning small datasets (CSV → JSON)
    - Writing small data transformation utilities using Scala

Outcome:

✅ Able to write concise, functional, and reusable Scala code for data manipulation

---

⚙️ Day 2 – Functional & Advanced Scala for ETL

Topics:

- Immutable vs Mutable collections
- Higher-Order Functions
- Currying and Partial Functions
- For-Comprehensions
- Anonymous and Lambda Functions
- Implicits and Type Parameters
- Object-Oriented Scala (Classes, Traits, Inheritance)
- Companion Objects and Apply methods
- Best Practices: Functional Programming for ETL
- Working with Files (read/write large files using Scala I/O)
- Integration with JSON, CSV, and configuration files
- Introduction to Futures and parallel collections (concurrent transformations)

Hands-on Labs:

- Create a Scala data cleansing utility with functional transformations
- Parallelize ETL transformations using Futures

Outcome:

✅ Understand advanced Scala features needed to write Spark ETL code

---

🚀 Day 3 – Hadoop & Spark Core with Scala

Topics:

- Big Data Ecosystem Overview: Hadoop, Hive, Spark, Kafka
- Hadoop Architecture: HDFS, YARN, Resource Manager
- Spark Architecture: Driver, Executors, Cluster Modes
- RDD Programming in Scala
    - Creating RDDs from files and HDFS
    - Transformations and Actions
    - Pair RDDs and Key-Value operations
    - Caching and Persistence
- Spark DataFrame API Basics
    - Schema Inference
    - DSL queries (`select`, `filter`, `groupBy`, `agg`, `join`)
    - UDFs in Scala
- Integrating Spark with Hadoop (read/write from HDFS)

Hands-on Labs:

- Read from HDFS → Transform → Write back using RDDs and DataFrames
- Simple ETL job using RDD and DataFrame APIs

Outcome:

✅ Build and run Spark jobs in Scala, integrated with Hadoop (HDFS)

🏗️ Day 4 – Spark SQL, Hive & ETL Pipeline Development

Topics:

- Spark SQL & Hive Integration
    - Working with SparkSession
    - Hive metastore configuration
    - Creating external tables
    - Querying Hive tables with Spark SQL
    - Schema evolution and partitioned tables
- ETL Design Patterns
    - Incremental Load
    - Merge / Upsert using Delta Lake or Spark SQL
    - Handling SCD (Slowly Changing Dimensions)
    - Data validation & error handling
- File Formats & Compression
    - Parquet, ORC, Avro — schema and performance
- Spark Job Optimization
    - Partitioning, Coalesce, Caching
    - Broadcast joins, Skew handling

Hands-on Labs:

- Build an end-to-end ETL pipeline:
  → Extract CSV from HDFS
  → Transform using Spark
  → Load to Hive external table (Parquet format)

- Query transformed data with Spark SQL

Outcome:

✅ Build real-world ETL pipelines in Spark + Hive using Scala

⚡ Day 5 – Advanced Spark, Tuning, and Project

Topics:

- Performance Tuning
    - DAG visualization, job stages, shuffle optimization
    - Executor memory, cores, and cluster config tuning
- Data Quality & Error Recovery
    - Handling bad records, retries, checkpointing
- Spark Structured Streaming (Intro)
    - Reading from Kafka / File Streams
    - Window operations and watermarking
- Testing and Deployment
    - Unit testing Spark jobs (ScalaTest)
    - Packaging with SBT & Spark Submit
    - Deploying to Hadoop / Kubernetes
- Overview of Modern ETL Frameworks
    - Delta Lake, Iceberg, Apache Hudi
    - Comparison: Spark vs Flink vs Beam

Final Capstone Project:

Mini Project: Build a Complete ETL Pipeline

1. Ingest raw JSON/CSV data from HDFS
2. Clean, deduplicate, and join with lookup data
3. Transform and store as Parquet in Hive table
4. Perform aggregations and analytics queries
5. Package and deploy with `spark-submit`

Outcome:

✅ End-to-end ETL job running on Spark + Hadoop + Hive, tuned and tested

---

🧠 Tools & Environment

Category	Tools / Versions
Language	Scala 2.12.15
Big Data	Hadoop 3.3.6, Hive 2.3.9
Engine	Spark 3.3.4 (Hadoop 3.3 prebuilt)
Build Tool	SBT or IntelliJ IDEA
Runtime	Java 11
Optional	Kafka (for streaming demo), Delta Lake 2.x

📦 Deliverables

- Full slide deck (concepts + examples)
- Hands-on notebooks / SBT projects

- Mini-project ETL pipeline (end-to-end)
- Cheat sheets (Scala syntax, Spark optimization)
- Assessment (MCQs + coding)
