📘 FINAL 50-DAY SYLLABUS

Java + Spring Boot (Project-Based Learning)

Project: School Management System (SMS)

🔹 DAY 1 – Project & Software Basics

What We Build
– School Management System flow diagram

How We Build
– Identify Admin, Teacher, Student
– Draw data flow (login → marks → fees → view result)

Why Necessary
– Coding without understanding leads to failure

When Used
– Requirement analysis phase

Student Learns
– What software is
– How systems work logically

🔹 DAY 2 – Java OOP Basics

What We Build
– Student & Teacher classes

How We Build
– class, object, variables, methods

Why Necessary
– Spring is built on Java OOP

When Used
– Every Java backend application

Student Learns
– Class, Object
– Encapsulation

🔹 DAY 3 – Why Spring & Spring Boot

What We Build
– First Spring Boot project

How We Build
– Spring Initializr
– Run main method

Why Necessary
– Reduces boilerplate & configuration

When Used
– Enterprise & startup apps

Student Learns
– Framework concept
– Spring vs Spring Boot

🔹 DAY 4 – Project Structure

What We Build
– Controller, Service, Repository layers

How We Build
– Package separation

Why Necessary
– Clean & maintainable code

When Used
– Industry projects

Student Learns
– Layered architecture

🔹 DAY 5 – Maven

What We Build
– Dependency management

How We Build
– pom.xml

Why Necessary
– Avoid manual JAR handling

When Used
– Build automation

Student Learns
– Maven basics

🔹 DAY 6 – Spring Bean

What We Build
– Spring-managed class

How We Build
– @Component

Why Necessary
– Spring controls object creation

When Used
– Dependency Injection

Student Learns
– Bean lifecycle

🔹 DAY 7 – Dependency Injection

What We Build
– Inject Service into Controller

How We Build
– @Autowired

Why Necessary
– Loose coupling

When Used
– Scalable applications

Student Learns
– DI concept

🔹 DAY 8 – Spring Annotations

What We Build
– REST Controller & Service

How We Build
– @RestController, @Service

Why Necessary
– Reduces boilerplate code

When Used
– Modern Spring apps

Student Learns
– Annotation-based config

🔹 DAY 9 – Request Flow

What We Build
– Controller → Service → Repository flow

How We Build
– Simple API

Why Necessary
– Understand backend working

When Used
– Debugging

Student Learns
– Request lifecycle

🔹 DAY 10 – Practice Day

What We Build
– Hello API

How We Build
– @GetMapping

Why Necessary
– Build confidence

When Used
– API testing

Student Learns
– REST basics

🔹 DAY 11 – REST API Basics

What We Build
– Student Controller

How We Build
– REST endpoints

Why Necessary
– Frontend communication

When Used
– Web & mobile apps

Student Learns
– REST architecture

🔹 DAY 12 – HTTP Methods

What We Build
– Add Student API

How We Build
– @PostMapping

Why Necessary
– Data creation

When Used
– Forms

Student Learns
– GET vs POST

🔹 DAY 13 – JSON & RequestBody

What We Build
– Accept student data

How We Build
– @RequestBody

Why Necessary
– Data exchange

When Used
– APIs

Student Learns
– JSON format

🔹 DAY 14 – Path Variable

What We Build
– Get student by ID

How We Build
– @PathVariable

Why Necessary
– Dynamic data access

When Used
– Search APIs

Student Learns
– URL parameters

🔹 DAY 15 – DTO

What We Build
– StudentDTO

How We Build
– Separate entity & response

Why Necessary
– Data security

When Used
– Production APIs

Student Learns
– DTO pattern

🔹 DAY 16 – Database Basics

What We Build
– MySQL connection

How We Build
– application.yml

Why Necessary
– Persistent storage

When Used
– All systems

Student Learns
– Database fundamentals

🔹 DAY 17 – JPA Entity

What We Build
– Student entity

How We Build
– @Entity

Why Necessary
– ORM mapping

When Used
– Database-driven apps

Student Learns
– Entity mapping

🔹 DAY 18 – Repository

What We Build
– StudentRepository

How We Build
– JpaRepository

Why Necessary
– Database operations

When Used
– CRUD

Student Learns
– Repository pattern

🔹 DAY 19 – CRUD Operations

What We Build
– Save & fetch students

How We Build
– Repository methods

Why Necessary
– Data handling

When Used
– Every application

Student Learns
– CRUD logic

🔹 DAY 20 – Update & Delete

What We Build
– Update & delete APIs

How We Build
– PUT & DELETE

Why Necessary
– Data maintenance

When Used
– Admin systems

Student Learns
– Full CRUD

🔹 DAY 21 – Teacher Entity

What We Build
– Teacher table

How We Build
– New entity

Why Necessary
– Multiple users

When Used
– School systems

Student Learns
– Multi-entity design

🔹 DAY 22 – Subject Entity

What We Build
– Subject table

How We Build
– Entity mapping

Why Necessary
– Subject-wise marks

When Used
– Education software

Student Learns
– Table design

🔹 DAY 23 – Relationships

What We Build
– Teacher–Subject mapping

How We Build
– @OneToMany

Why Necessary
– Data linking

When Used
– Relational DBs

Student Learns
– Relationships

🔹 DAY 24 – Marks Entity

What We Build
– Marks table

How We Build
– Join relationships

Why Necessary
– Academic records

When Used
– Marksheet systems

Student Learns
– Join operations

🔹 DAY 25 – Marksheet

What We Build
– Marksheet API

How We Build
– Combine marks

Why Necessary
– Result display

When Used
– Schools

Student Learns
– Data aggregation

🔹 DAY 26 – Fees Module

What We Build
– Fees entity

How We Build
– Paid/unpaid logic

Why Necessary
– Business rules

When Used
– Paid systems

Student Learns
– Business logic

🔹 DAY 27 – Validation

What We Build
– Input validation

How We Build
– @NotNull, @Email

Why Necessary
– Prevent bad data

When Used
– Forms & APIs

Student Learns
– Validation rules

🔹 DAY 28 – Exception Handling

What We Build
– Global exception handler

How We Build
– @ControllerAdvice

Why Necessary
– Proper error handling

When Used
– Production apps

Student Learns
– Exception flow

🔹 DAY 29 – Fees-Based Access

What We Build
– Restrict marksheet access

How We Build
– Conditional logic

Why Necessary
– Business enforcement

When Used
– Subscription systems

Student Learns
– Conditional access

🔹 DAY 30 – Mid Review

What We Build
– Test full flow

How We Build
– Postman

Why Necessary
– Stability

When Used
– QA phase

Student Learns
– Debugging

🔹 DAY 31 – Security Basics

What We Build
– Add Spring Security

How We Build
– Dependency & config

Why Necessary
– Protect data

When Used
– Secure apps

Student Learns
– Security basics

🔹 DAY 32 – Authentication

What We Build
– Login system

How We Build
– UserDetailsService

Why Necessary
– Identity check

When Used
– All apps

Student Learns
– Authentication

🔹 DAY 33 – Password Encryption

What We Build
– Encrypted passwords

How We Build
– BCrypt

Why Necessary
– Password safety

When Used
– User systems

Student Learns
– Encryption basics

🔹 DAY 34 – Roles

What We Build
– Admin, Teacher, Student roles

How We Build
– Role mapping

Why Necessary
– Access control

When Used
– Multi-user systems

Student Learns
– Authorization

🔹 DAY 35 – Role-Based APIs

What We Build
– Teacher-only marks entry

How We Build
– Security config

Why Necessary
– Data protection

When Used
– Secure systems

Student Learns
– Role restrictions

🔹 DAY 36 – JWT

What We Build
– Token generation

How We Build
– JWT utility

Why Necessary
– Stateless auth

When Used
– REST APIs

Student Learns
– JWT concept

🔹 DAY 37 – JWT Filter

What We Build
– Token validation

How We Build
– Security filter

Why Necessary
– Request authentication

When Used
– Secure APIs

Student Learns
– Token flow

🔹 DAY 38 – Secure Marksheet

What We Build
– Fees + role + JWT check

How We Build
– Combined logic

Why Necessary
– Business security

When Used
– Paid platforms

Student Learns
– Real security rules

🔹 DAY 39 – Method Security

What We Build
– Secure methods

How We Build
– @PreAuthorize

Why Necessary
– Fine control

When Used
– Enterprise apps

Student Learns
– Method security

🔹 DAY 40 – Security Testing

What We Build
– Secure API testing

How We Build
– Postman

Why Necessary
– Verify protection

When Used
– QA

Student Learns
– Security debugging

🔹 DAY 41 – Logging

What We Build
– Application logs

How We Build
– Logger

Why Necessary
– Debug issues

When Used
– Production

Student Learns
– Logging

🔹 DAY 42 – Swagger

What We Build
– API documentation

How We Build
– Swagger config

Why Necessary
– API clarity

When Used
– Team projects

Student Learns
– API docs

🔹 DAY 43 – Actuator

What We Build
– Health endpoint

How We Build
– Actuator

Why Necessary
– Monitoring

When Used
– Live apps

Student Learns
– Monitoring

🔹 DAY 44 – AOP

What We Build
– Logging aspect

How We Build
– @Aspect

Why Necessary
– Separate concerns

When Used
– Large projects

Student Learns
– AOP basics

🔹 DAY 45 – Auditing

What We Build
– Created & updated time

How We Build
– Auditing annotations

Why Necessary
– Track changes

When Used
– Databases

Student Learns
– Auditing

🔹 DAY 46 – Testing Basics

What We Build
– JUnit test

How We Build
– @Test

Why Necessary
– Bug prevention

When Used
– Development

Student Learns
– Testing basics

🔹 DAY 47 – Service Testing

What We Build
– Service unit test

How We Build
– Mockito

Why Necessary
– Logic testing

When Used
– CI/CD

Student Learns
– Mocking

🔹 DAY 48 – Integration Testing

What We Build
– API testing

How We Build
– Full flow tests

Why Necessary
– Reliability

When Used
– Pre-release

Student Learns
– End-to-end testing

🔹 DAY 49 – Final Review

What We Build
– Clean & refactor code

How We Build
– Best practices

Why Necessary
– Maintainability

When Used
– Long-term projects

Student Learns
– Clean code

🔹 DAY 50 – Final Demo & Viva

What We Build
– Complete SMS demo

How We Build
– Explain architecture & flow

Why Necessary
– Confidence & clarity

When Used
– Interviews

Student Learns
– Project explanation
– Interview readiness