# **Phase 8: Data Management & Deployment Report**

## **VS Code & SFDX Integration Steps**

### **1. Environment Setup**

* Installed Visual Studio Code
* Installed Salesforce Extension Pack
* Configured SFDX CLI (Salesforce DX Command Line Interface)
* Verified SFDX installation using sfdx --version

### **2. Project Initialization**

* Created new SFDX project using sfdx project:create
* Configured project structure with standard directories:
  + /force-app/main/default/classes/
  + /force-app/main/default/triggers/
  + /force-app/main/default/objects/

### **3. Salesforce Org Connection**

* Authenticated with Salesforce org using sfdx auth:web:login
* Set default org for development
* Verified connection and permissions

### **4. Class Development**

* Created Apex classes in VS Code
* Utilized IntelliSense for code completion
* Implemented proper error handling and best practices
* Added comprehensive code comments and documentation

### **5. Code Synchronization**

* Used sfdx source:push to deploy classes to scratch org
* Retrieved metadata using sfdx source:pull
* Maintained version control with Git integration

### **6. Testing & Validation**

* Wrote unit tests for all custom classes
* Executed tests using sfdx apex:test:run
* Verified code coverage requirements (minimum 75%)
* Resolved any compilation or runtime errors

### **7. Deployment Preparation**

* Validated deployment using sfdx source:deploy --checkonly
* Created deployment packages
* Generated change sets for production deployment
* Documented deployment dependencies

### **8. Data Management**

* Exported sample data using SFDX data commands
* Created data import/export scripts
* Managed test data for different environments
* Implemented data backup strategies

## **Key Deliverables**

* ✅ Functional Apex classes deployed to org
* ✅ Unit tests with adequate code coverage
* ✅ Version-controlled codebase
* ✅ Deployment documentation
* ✅ Data management procedures

