**Tasks – DTN ADMLOD Export Prototype**

**Prepared by: Jordan Eisenback  
Date: October 5, 2025**

**✅ Completed Tasks**

**✅ Project Setup**

* **Initialized C# .NET console application repository.**
* **Created directory structure including /MockFiles.**
* **Added README with build/run instructions.**

**✅ ADMLOD Export File Generation**

* **Implemented AdmLodFileGenerator.cs to generate ADMLOD export files.**
* **Included correct header, export command, and MERGE lines.**
* **Saved output to /MockFiles/adm\_export.txt.**

**✅ FTP Upload Simulation**

* **Implemented FtpClientMock.cs to simulate FTP upload.**
* **Logged upload action.**
* **Added logic to simulate FTP failure via CLI flag.**

**✅ Response Parsing**

* **Implemented ResponseParser.cs to parse DTN response files.**
* **Displayed success and error counts.**
* **Logged parsing results.**

**✅ Logging**

* **Implemented Logger.cs for console and file logging.**
* **Included severity, timestamps, and correlation IDs.**
* **Supported --logfile CLI argument.**

**✅ CLI Argument Handling**

* **Implemented CLI parsing in Program.cs.**
* **Supported export, --success, --error, and --logfile.**
* **Handled invalid or missing arguments gracefully.**

**✅ Error Scenario Simulation**

* **Simulated and logged:** 
  + **Missing required fields**
  + **FTP upload failure**
  + **Invalid command format**
  + **Errors in DTN response file**

**✅ Output Review and Acceptance**

* **Verified ADMLOD format compliance.**
* **Confirmed response output clarity.**
* **Demonstrated both success and error scenarios.**

**🔜 Upcoming Tasks**

**🔜 Mocked Persistence Layer**

* **Replace MockData.cs with:** 
  + **IProductRepository and JsonProductRepository.cs**
  + **IProductGroupRepository and JsonGroupRepository.cs**
* **Load data from products.json and productgroups.json.**
* **Include error cases for validation testing.**

**🔜 Dynamic DTN Response Generation**

* **Define IResponseGenerator interface.**
* **Implement MockResponseGenerator.cs to generate responses based on export command validation.**
* **Replace static response files with dynamic logic.**

**🔜 Dependency Injection Setup**

* **Refactor Program.cs to inject:** 
  + **Repositories**
  + **Response generator**
  + **Logger**
* **Ensure all services are testable and mockable.**

**🔜 Configuration Management**

* **Create appsettings.json for:** 
  + **File paths**
  + **Logging options**
  + **Feature flags**
* **Load configuration via strongly typed classes.**

**🔜 Documentation**

* **Update README with:** 
  + **Repository and response generator usage**
  + **Configuration file structure**
  + **DI setup instructions**

**🔜 Testing**

* **Expand test plan to cover:** 
  + **Repository-driven data scenarios**
  + **Dynamic response validation**
  + **DI behavior and service substitution**
* **Log results and defects.**
* **Verify acceptance criteria for updated user stories.**