**Scan event exercise:**

**Indika Kamal**

**Requirements**

1. The application must consume from a scan event API and keep a track of the last event that was fetched so that it can continue where it left off if the service is stopped and started.
2. The application should persist the scan event data in such a way that the following information could later be fetched easily;
3. The most recent scan event against a parcel, specifically only values from fields; EventId, ParcelId, Type, CreatedDateTimeUtc, StatusCode, RunId are required.
4. DateTimes indicating when a parcel has been;
   1. Picked up (a scan event of Type ‘PICKUP’ has occurred).
   2. Delivered (a scan event of Type ‘DELIVERY’ has occurred).
5. The application should be fault tolerant and resilient (e.g. handle new event types, malformed data).
6. The application should contain appropriate logging.

Assumptions

1. List any assumptions you have made. If anything in these instructions is not clear, write down an assumption that captures this.

Improvements

1. List any improvements that could be made to the application you have built including what kind of things could be done to productionise this application.
2. Describe what kind of changes or additions could be performed to enable another worker application downstream to also perform actions against the same scan events processed by this application. Describe what the high-level architecture of this overall system could look like.

**Answers:**

Used Libraries:

* FluentValidation for validation
* Serilog – for logging

Assumptions:

* 1. API send data in descending order based on most recent records (EventID)
  2. DB Layer has not been implemented to save cleanup event data

Improvements:

* Entity Framework can use to save data to database in order to show in client application
* Configure to install as a Service in the server
* Web API is required to implement

Completed:

* Created the Worker App based on given requirements
* Applied Validation using Fluent Validation
* Serilog is used for events and error Logging