Log with Serilog

Creating logger in program.cs

```
Log.Logger = new LoggerConfiguration()
.MinimumLevel.Information()
.WriteTo.Console()
.WriteTo.File("logs/log.txt", rollingInterval: RollingInterval.Day)
.CreateLogger();
```

```
//towrite in seperate file and
Log.Logger = new LoggerConfiguration()
    .MinimumLevel.Information()
    .WriteTo.Console()
    .WriteTo.Logger(
        x => x.Filter.ByIncludingOnly(e => e.Level == LogEventLevel.Error)
            .WriteTo.File($"Logs/Error.txt", rollingInterval: RollingInterval.Day)
    )
    .WriteTo.Logger(
        x => x.Filter.ByIncludingOnly(e => e.Level == LogEventLevel.Warning)
            .WriteTo.File($"Logs/Warning.txt", rollingInterval: RollingInterval.Day)
    )
    .WriteTo.Logger(
        x => x.Filter.ByIncludingOnly(e => e.Level == LogEventLevel.Information)
            .WriteTo.File($"Logs/Information.txt", rollingInterval:
RollingInterval.Day)
    )
    .WriteTo.File($"Logs/log.txt", rollingInterval: RollingInterval.Day)
    .WriteTo.Console()
    .CreateLogger();
```

Level System

- Fatal - something that will result in a crash
- Error - serious problem encountered
- Warning - to report non crucial information
- Information - to inform
- Debug
- Verbose

Invoking logs

```
Log.Information("Token Generated for {@UserId} with {@token} ", userDetail.Id,
userCredential.Token);
Log.Error(messageTemplate: "Unable to delete product {@Id} \\n {@message}",
command.ProductId, ex.Message);
Log.Warning("Invalid Credentials: {@credentials}", credentials);
```