Requirements

- Receive payload.
- Store main information, including temperature, humidity, and occupancy.
- Handle exceptions.
- Implement logging to capture application events and errors.
- Maintain clean code practices.

Technologies Used

- .NET Core
- Entity Framework Core (Db First)
- Serilog (for logging)
- MSSQL Database

Project Structure

```
- Controllers/
    - PayloadController.cs
- Exceptions/
    - DatabaseOperationException.cs
   - PayloadValidationException.cs
- Middlewares/
   - ErrorHandlingMiddleware.cs
   - LoggingMiddleware.cs
- Context/
   - AppDbContext.cs
- Models/
   - Payload.cs
-RequestModels/
  -PayLoadModel
```

- Repositories/
  - IPayLoadRepository.cs
  - PayLoadRepository.cs

- Services/
  - IPayLoadService.cs
  - PayLoadService.cs
- appsettings.json
- Program.cs

Configuration

- Database Connection String: Update the **appsettings.json** file to specify the connection string for your chosen database backend.
- Database Script: **DatabaseScriptForDotNetCoreApiProject.sql**

Usage

Endpoint: /api/payloads

- Description: Submit payload data
- Method: POST
- Request Body: JSON payload
- Example Request:

```
{
"deviceId": "ibm-878A66",
"deviceType": "computer1.0.0",
"deviceName": "VN1-1-3",
"groupId": "847b3b2f1b05dc4",
"dataType": "DATA",
"data": {
  "fullPowerMode": false,
  "activePowerControl": false,
  "firmwareVersion": 1,
    "temperature": 21,
    "humidity": 53
  },
  "timestamp": 1629369697
}
```

- Example Response:

```
{
  "message": "Payload data received and stored successfully."
}
```

Logging

- Create log file as log.txt under project directory and save system information and errors