**Develop a Dynamic Mapping System Using .NET**

For this task a .NET API project is created.

**System Architecture**

* A main program file is there for execution and calling all the defined classes and functions.
* Project consists of main folders Model and Utilities.
* Model consists of required classes for the structure of data. Can be extended with additional model classes when required. As a sample Reservation Model is added for testing.
* Utilities consists of the MapHandler class which handles the mapping of a model between JSON and XML. Additional types can be added. The methods are defined as generics so any model can be passed.
* Utilities also has a custom error handler class which can handle any error thrown in the program and show the error message or status code as output.

**Testing and additional details**

* Inside the program itself XML and JSON conversion is tested with sample endpoints :
  + <http://localhost:5194/reservation/json-to-xml>
  + <http://localhost:5194/reservation/json-to-xml>
* For extensive testing another endpoint as following is created:
  + <http://localhost:5194/reservation/convert?sourceType=JSON&targetType=XML>
  + This is for JSON to XML.
  + This requires the JSON data:

{

"Reservation": {

"Id": 123,

"Name": "John",

"BookingType": "Room",

"Amount": 32

}

}

* + <http://localhost:5194/reservation/convert?sourceType=XML&targetType=JSON>
  + This is for XML to JSON
  + This requires the XML data:

<ReservationCollectionModel>

<Reservation>

<Id>123</Id>

<Name>John</Name>

<BookingType>Room</BookingType>

<Amount>32</Amount>

</Reservation>

</ReservationCollectionModel>

* + <http://localhost:5194/reservation/convert?sourceType=XML&targetType=Test>
  + This is for error handling testing.
* The testing is done using Postman.

**Assumptions and limitations**

* The program should work with Models instead of directly converting between JSON and XML(for Ex using NewtonSoft.JSON for converting directly).
* Additional entries of elements or properties not mapped or not available in the model will be ignored.