**Answer the following questions with your submission to Brightspace (as text submission): 1. Define each HTTP Status Code that you used in this project and explain what each code**

1. Status code 200 [OK]:
   * The request has been successful, and the server returns the data requested.
   * This status code is typically used when retrieving a list of students or a specific student. It indicates that the operation was successful, and the server is sending back the requested data.
2. Status code 201 [Created]:
   * A 201 status code means that a new resource has been created.
   * For example, when adding a new student using the POST action, a 201 status response means the student has been successfully added as a new resource in the system.
3. Status code 204 [No Content]:
   * The request has been fully processed, but there is no data to return.
   * This status code is often used for DELETE actions, showing that the deletion operation was successful, but there is no additional content to send back to the client.
4. Status code 400 [Bad Request]:
   * The server cannot or will not process the request due to a client error.
   * This status code is returned when the request is invalid or missing parameters. It indicates that the server cannot understand or process the request due to a client-side mistake.
5. Status code 500 [Internal Server Error]:
   * The server encountered an error while processing the request.
   * This status code indicates that there was an unexpected internal error on the server while handling the request. It signifies that the server failed to fulfill the request due to an issue on the server-side.

Does the service you created in this assignment conform to all REST principals? Explain why.

Summary:

The service created in this assignment partially adheres to REST principles by successfully implementing the fundamental CRUD operations (Create, Read, Update, Delete) for the 'Student' resource using the appropriate HTTP methods (POST, GET, PUT, DELETE). However, it lacks support for the 'PATCH' method, which allows partial updates, and could benefit from consistent usage of appropriate HTTP status codes and improved response bodies. To achieve full REST conformance, the service should incorporate 'PATCH' support to enable partial modifications, consistently utilize relevant status codes to provide clear feedback on request outcomes, and enhance response bodies with pertinent data for successful operations and meaningful error messages for error scenarios, leading to a more comprehensive and standardized API experience for clients.