Requirements Shared:

End Point(s):

API Endpoint	/Prod/next-birthday
Method	GET
Input	• Dateofbirth → type:date → Accepts in YYYY-MM-DD format • Unit → type:string → Acceptable units hour/day/week/month
Output	Response: A json string Containing the unit measure of the time left till the next birthday
Purpose	API endpoint that calculates the amount of time left before user's next birthday.

Test Strategy:

Overview: We want to be able to run regression tests for this endpoint automatically, create a test suite with all necessary tests that we need to execute to be sure next-birthday endpoint works correctly.

Scope:

Regression	Mandatory
Performance	Optional (High
Out of Scope	Unit testing

Test Environment: Production (API is openly available for public to access at this point)

Testing Tools: Robot + Python Framework

Deliverables:

- Bugs Report
- link to tests' code (e.g GitHub, Bitbucket etc.) with instructions to run tests
- Observations
- Enhancements if any

Timeline:

Testing Start Date	2023-06-10 (Tentative)
Testing End Date	2023-06-11

Approvals:
Not Required at this time

Release Date: N/A

Bug 1: endpoint should ignore the case for units.
Priority: low
Description: When I do a get call on the endpoint (/Prod/next-birthday) with unit set to Hour it returns a 200 response with null as message.
Expectation: Endpoint should honor units irrespective of case and also respond with an appropriate error message.

Bug2: endpoint should display appropriate message for when one of the arguments is missing
Priority: High
Description: When I do a get call on the endpoint (/Prod/next-birthday) with unit set to hour, while date birth of is unset, endpoint returns a 200 response with "Please specify both query parameter dateofbirth and unit" as message.
Expectation:
Endpoint should not respond with a 200 OK and instead should respond with a 400 error with appropriate message containing details of the missing variable

Bug 3: endpoint should respond with appropriate error codes and messages.
Priority: High
Description: When I do a get call on the endpoint (/Prod/next-birthday) with unit set to Hour it returns a 200 response with null as message.
Expectation:
Endpoint should not respond with a 200 OK and instead should respond with 400 error with appropriate message

Bug 4: endpoint should respond with 400 bad request for invalid inputs.
Priority: High
Description: When I do a get call on the endpoint (/Prod/next-birthday) with unit set to seconds (which is not supported) it returns a 200 response with null as message.
Expectation: Endpoint should not respond with a 200 OK and instead should respond with 400 error with appropriate message.

Enhancements:
Enhancement-1: endpoint should respond with appropriate message when the birthdate is closer than unit
Priority: Medium
Description: When I do a get call on the endpoint (/Prod/next-birthday) with unit set to Months for a birthday within the same month and the unit is set to months it returns a 200 response with 0 months left as message.
Expectation:
Endpoint should respond with an appropriate message such as "Less than a month left"

Enhancement-2: endpoint can ignore the year from inputs.
Priority: low
Description: Since we are only concerned with the time distance from the birthdate and there are no age related interactions, we can safely ignore the year from the inputs and as users to send the date in yy-mm format

Test Status:

Testing Start Date	2023-06-11 (Actual)
Testing End Date	2023-06-11
Regression Test Cases	11
Performance Tests	1
Bugs	4
Enhancements	2
Total Time spent on the testing Activity	6

Observations:
There was no access to backend which limited the availability to run E2E. Only the existing behavior was specified so the verification and validation are based on the existing behavior alone, which is partial regression. In order to run a complete regression, information about the previous behavior of the API and the list of changes also need to be specified.

Since there was no access to the code, the ability to validate response code was also limited to 200 and 403 statuses only. Access to code and the backend could've enabled us to validate 3xx, 4xx and 5xx response codes as well along with appropriate messages.