Task#2:

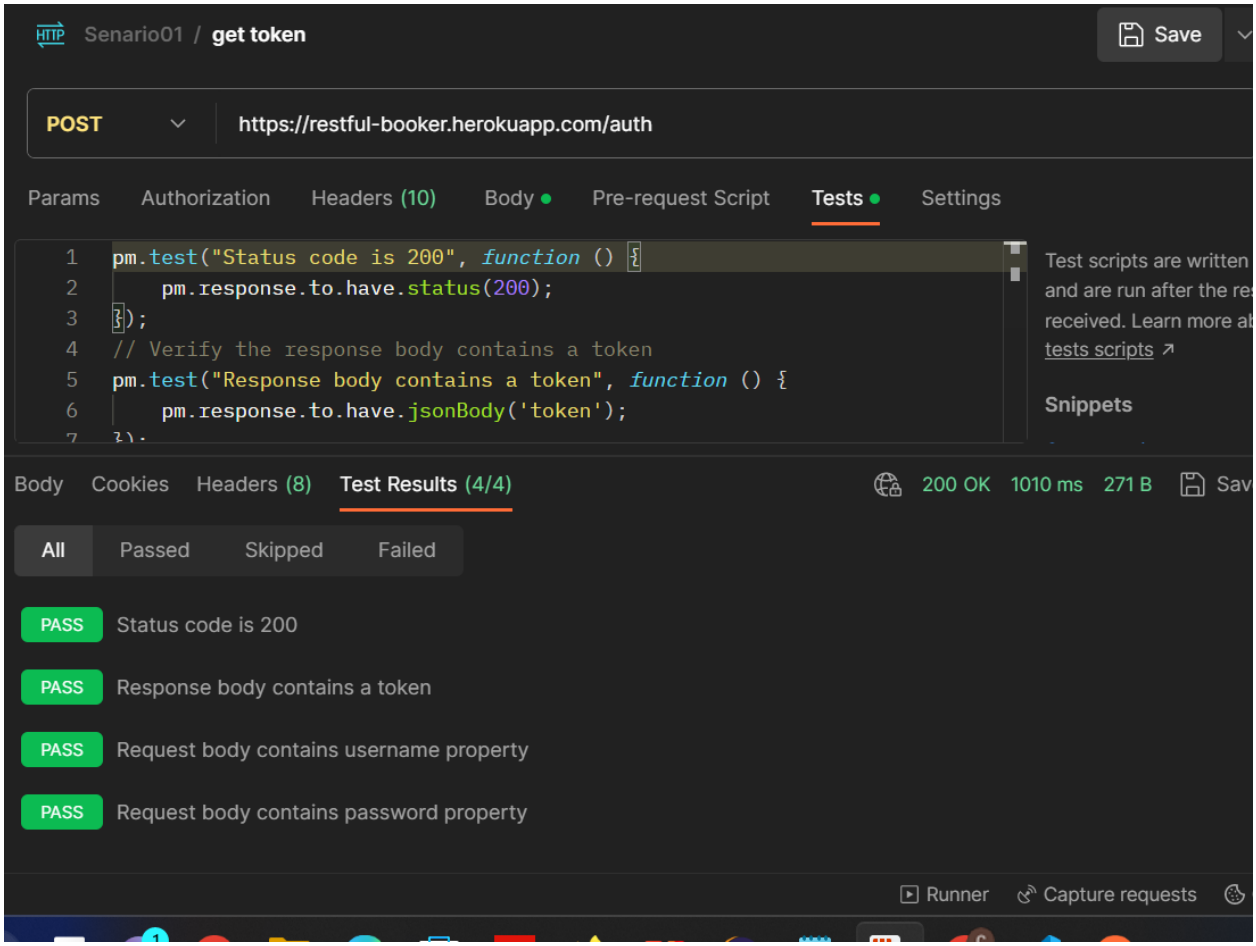
Given the API specification documentation attached above, create an API test automation
suite using any tool of your choice to cover all positive and negative scenarios of the CreateBooking endpoint specified in the API documentation.

(API testing Code is Shared to GitHub Repository)

https://github.com/chamodya-layaji97/QA_Engineer_PracticalTest.git

1.Get Token Test

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
// Verify the response body contains a token
pm.test("Response body contains a token", function () {
    pm.response.to.have.jsonBody('token');
});
pm.test("Request body contains username property", function () {
    var requestBody = JSON.parse(pm.request.body.raw);
    pm.expect(requestBody).to.have.property("username");
});
pm.test("Request body contains password property", function () {
    var requestBody = JSON.parse(pm.request.body.raw);
    pm.expect(requestBody).to.have.property("password");
});
```

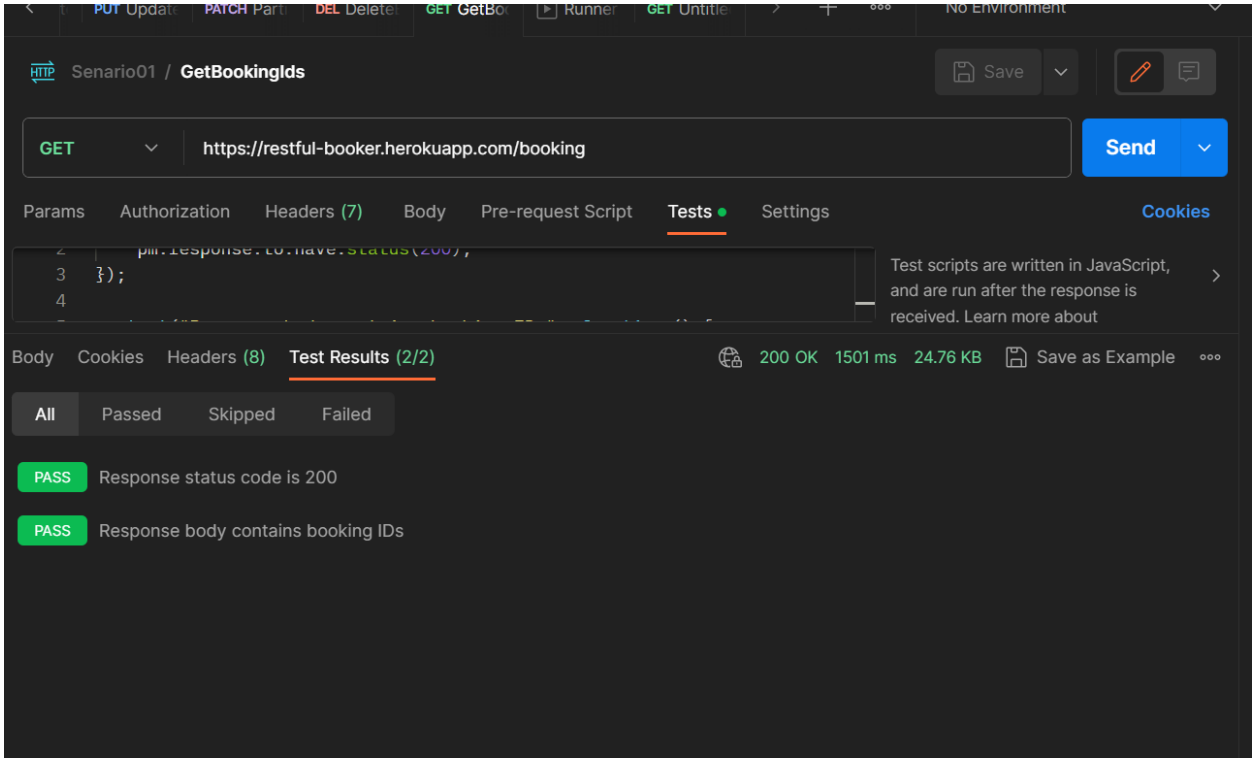


2.GetBookingIds

```
pm.test("Response status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Response body contains booking IDs", function () {
    var responseBody = pm.response.json();
    pm.expect(responseBody).to.be.an('array').that.is.not.empty;

    for (var i = 0; i < responseBody.length; i++) {
        pm.expect(responseBody[i]).to.have.property('bookingid');
        pm.expect(responseBody[i].bookingid).to.be.a('number');
    }
});
```



3.GetBooking

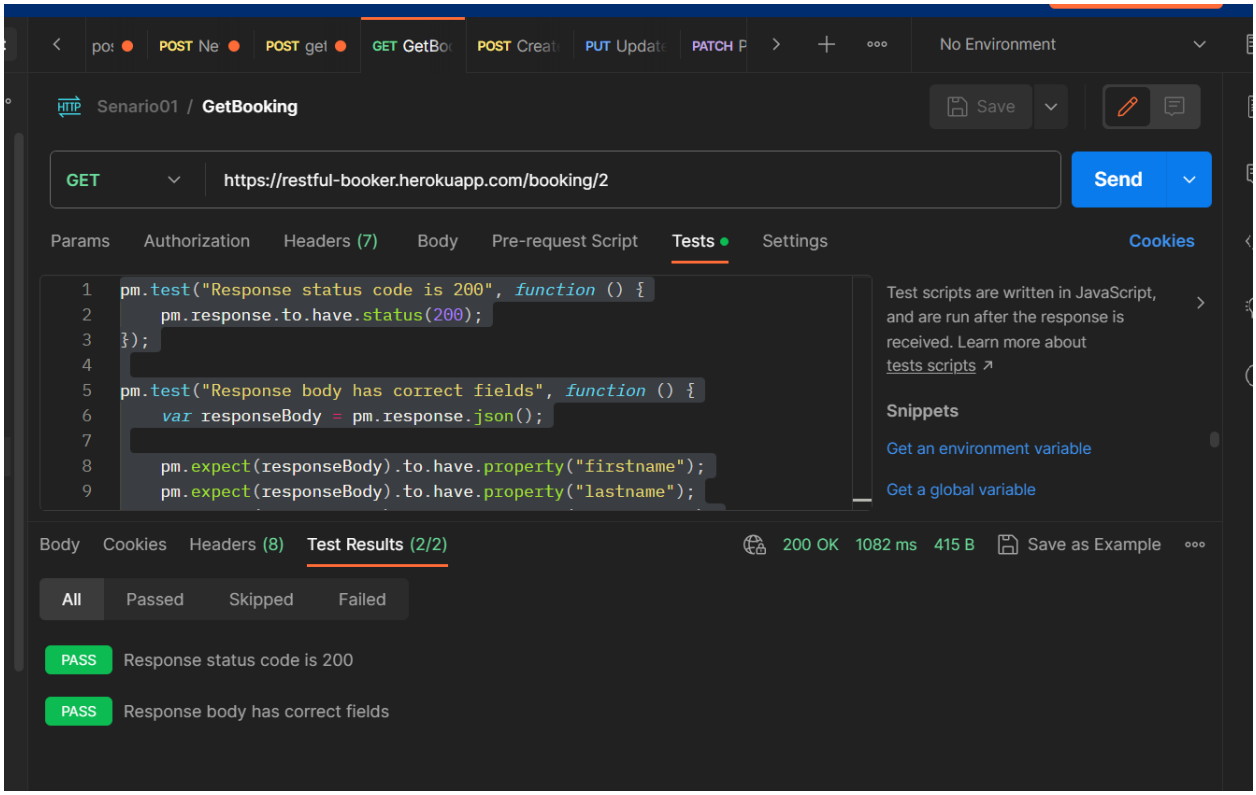
```
pm.test("Response status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Response body has correct fields", function () {
    var responseBody = pm.response.json();

    pm.expect(responseBody).to.have.property("firstname");
    pm.expect(responseBody).to.have.property("lastname");
    pm.expect(responseBody).to.have.property("totalprice");
    pm.expect(responseBody).to.have.property("depositpaid");
    pm.expect(responseBody).to.have.property("bookingdates");
    pm.expect(responseBody).to.have.property("additionalneeds");

    pm.expect(responseBody.firstname).to.be.a("string");
    pm.expect(responseBody.lastname).to.be.a("string");
    pm.expect(responseBody.totalprice).to.be.a("number");
    pm.expect(responseBody.depositpaid).to.be.a("boolean");
```

```
    pm.expect(responseBody.bookingdates).to.be.an("object");
    pm.expect(responseBody.additionalneeds).to.be.a("string");
});
```



4.CreateBooking

```
pm.test("Response status code is 200", function () {
    pm.response.to.have.status(200);
});

pm.test("Response body has booking ID and details", function () {
    var responseBody = pm.response.json();

    pm.expect(responseBody).to.have.property("bookingid");
    pm.expect(responseBody.bookingid).to.be.a("number");
    pm.expect(responseBody).to.have.property("booking").that.is.an("object");
```

```
    var booking = responseBody.booking;
    pm.expect(booking).to.have.property("firstname").that.is.a("string");
    pm.expect(booking).to.have.property("lastname").that.is.a("string");
    pm.expect(booking).to.have.property("totalprice").that.is.a("number");
    pm.expect(booking).to.have.property("depositpaid").that.is.a("boolean");
    pm.expect(booking).to.have.property("bookingdates").that.is.an("object");
    pm.expect(booking).to.have.property("additionalneeds").that.is.a("string");

    var bookingDates = booking.bookingdates;
    pm.expect(bookingDates).to.have.property("checkin").that.is.a("string");
    pm.expect(bookingDates).to.have.property("checkout").that.is.a("string");
});

// Positive Scenario - Successful Booking Creation
pm.test("Create Booking - Successful", function () {
    // Verify the response status code is 200
    pm.response.to.have.status(200);

    // Verify the response body contains the booking ID and details
    pm.expect(pm.response.json()).to.have.property("bookingid");
    pm.expect(pm.response.json().bookingid).to.be.a("number");
    pm.expect(pm.response.json()).to.have.property("booking").that.is.an("object");
});

// Negative Scenario - Missing Required Field
pm.test("Create Booking - Missing Required Field", function () {
    // Verify the response status code is 400
    pm.response.to.have.status(400);

    // Verify the response body contains the error message for the missing field
    pm.expect(pm.response.json().error).to.equal("Missing required field(s)");
});

// Negative Scenario - Invalid Data Type
pm.test("Create Booking - Invalid Data Type", function () {
    // Verify the response status code is 400
    pm.response.to.have.status(400);

    // Verify the response body contains the error message for the invalid data type
    pm.expect(pm.response.json().error).to.equal("Invalid data type");
});
```

```
// Negative Scenario - Invalid Date Format
pm.test("Create Booking - Invalid Date Format", function () {
    // Verify the response status code is 400
    pm.response.to.have.status(400);

    // Verify the response body contains the error message for the invalid date format
    pm.expect(pm.response.json().error).to.equal("Invalid date format");
});
```