Test Case Template for Project	
Project Name	Restful Broker
Module Name	CRUD Operation
Created By	Vanessa
Created Date	13-July-2022
Reviewer	Roy Dutta

TestCaseID	Scenario	METHOD	Scenario Description	TestCase Description	PreCond	TestSteps	Expected Result	Actual Result	Execution St	PostMan Scripts	Executed QA Name

TC01	PING HEALTH CHECK		Status Code	Verify if you are getting status code of 201 for ping health check		1.Login to POSTMAN tool
						2.In POSTMAN tool,Navigate to My workspace-->Collection->Click Addnew request
						3.Enter the PingHealth check url https://restful-booker.herokuapp.com/ping and Select GET request and Click Send button
						4.Verify if you are getting status code of 201 in response tab
						
						Scripting:
						pm.test("Status code is 201", function () {
						   pm.response.to.have.status(201);
						});
							1.Successful Login
							2.Successful Navigation
							3.The user must be able to send request
							4.The Status code in the Response tab must be 201
										Scripting:
										pm.test("Status code is 201", function () {
										   pm.response.to.have.status(201);
										});
											Vanessa

TC02	PING HEALTH CHECK		Response String	Verify if you are getting response body string as Created for ping health check		1.Login to POSTMAN tool
						2.In POSTMAN tool,Navigate to My workspace-->Collection->Click Addnew request
						3.Enter the PingHealth check url
						https://restful-booker.herokuapp.com/ping
						and Select GET request and Click Send button
						4.Verify if you are getting response body string as 'Created'
						
						Scripting:
						pm.test("Body matches string", function () {
						   pm.expect(pm.response.text()).to.include("Created");
						});
							1.Successful Login
							2.Successful Navigation
							3.The user must be able to send request
							4.The Response body must show string as 'Created'
										Scripting:
										pm.test("Body matches string", function ()
										   pm.expect(pm.response.text()).to.includ
										});
											Vanessa

TC03	PING HEALTH CHECK		Response	Verify if you are getting response as 201 Created		1.Login to POSTMAN tool
						2.In POSTMAN tool,Navigate to My workspace-->Collection->Click Addnew request
						3.Enter the PingHealth check url
						https://restful-booker.herokuapp.com/ping
						and Select GET request and Click Send button
						4.Verify if you are getting response body string as 'Created'
						
						Scripting:
						pm.test("Status code is 201", function () {
						   pm.response.to.have.status(201);
						});
						
						pm.test("Body matches string", function () {
						   pm.expect(pm.response.text()).to.include("Created");
						});
						
						pm.test("Content-Type is present", function () {
						   pm.response.to.have.header("Content-Type");
						});
							1.Successful Login
							2.Successful Navigation
							3.The user must be able to send request
							4.The Response body must show string as 'Created'
										Scripting:
										pm.test("Status code is 201", function () {
										   pm.response.to.have.status(201);
										});
										
										pm.test("Body matches string", function ()
										   pm.expect(pm.response.text()).to.includ
										});
										
										pm.test("Content-Type is present", functio
										   pm.response.to.have.header("Content-T
										});
											Vanessa

TC04	PING HEALTH CHECK		Response Header Content Type	Verify if Response Header has content type		1.Login to POSTMAN tool 2.In POSTMAN tool,Navigate to My workspace-->Collection->Click Addnew request 3.Enter the PingHealth check url https://restful-booker.herokuapp.com/ping and Select GET request and Click Send button 4.Verify if Response Header has content type **Scripting:** **pm.test("Content-Type is present", function () {** **pm.response.to.have.header("Content-Type");** **});**	1.Successful Login 2.Successful Navigation 3.The user must be able to send request 4.The Response header must have a content type			Scripting: pm.test("Content-Type is present", functio pm.response.to.have.header("Content-T });	Vanessa
											Vanessa
TC05	Get All Booking			Verify if Response time is lessthan 200ms		1.Login to POSTMAN tool 2.In POSTMan tool ,Import the Curl command using Raw text formal curl -i https://restful-booker.herokuapp.com/booking 3.Click send request button and Verify if Response time in less than 200ms **Scripting:** **pm.test("Response time is less than 200ms", function () {** **pm.expect(pm.response.responseTime).to.be.below(200);** **});**	1.Successful Login 2.Successful Navigation 3.Successful action and The Response time must be less than 200ms			Scripting: pm.test("Response time is less than 200m pm.expect(pm.response.responseTime). });	Vanessa

TC06	Get All Booking	GET		Verify if User is able to Fetch All bookings successfully		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text formal
						curl -i https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the response body
						
						Scripting:
						pm.test("Body matches string", function () {
						pm.expect(pm.response.text()).to.include("bookingid");
						});
							1.Successful Login
							2.Successful Navigation
							3.The API must fetch all bookings
										Scripting:
										pm.test("Body matches string", function ()
										pm.expect(pm.response.text()).to.includ
										});
											Vanessa

TC07	Get All Booking			Verify JSON Response for Get all Booking		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text formal
						curl -i https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the response body
						
						Scripting:
						pm.test("Body matches string", function () {
						pm.expect(pm.response.text()).to.include("bookingid");
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response body must show list of all bookings
										Scripting:
										pm.test("Body matches string", function ()
										pm.expect(pm.response.text()).to.includ
										});
											Vanessa

TC08	Get All Booking			Verify Get All Booking status code When all bookings are deleted		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text formal
						curl -i https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the status code when all bookings are deleted
						
						pm.test("Status code is 404", function () {
						pm.response.to.have.status(404);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Status code must be 404 for all booking deleted
										pm.test("Status code is 404", function () {
										pm.response.to.have.status(404);
										});
											Vanessa

TC09	Get All Booking			Verify Status code for Get All bookings when all booking id is fetched		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text formal
						curl -i https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the status code when all bookings are fetched successfully
						Scripting
						pm.test("Status code is 200", function () {
						pm.response.to.have.status(200);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Status code must be 200 for all booking deleted
										Scripting
										pm.test("Status code is 200", function () {
										pm.response.to.have.status(200);
										});
											Vanessa

TC10	Get Single Booking	GET	Response time	Verify if Response time is lessthan 200ms		1.Login to POSTMAN tool 2.In POSTMan tool ,Import the Curl command using Raw text formal curl -i https://restful-booker.herokuapp.com/booking/x x-->x can be your preferred booking id 3.Click send request button and Verify if Response time in less than 200ms **Scripting:** **pm.test("Response time is less than 200ms", function () {** **pm.expect(pm.response.responseTime).to.be.below(200);** **});**	1.Successful Login 2.Successful Navigation 3.Successful action and The Response time must be less than 200ms			Scripting: pm.test("Response time is less than 200m pm.expect(pm.response.responseTime). });	Vanessa
TC11				Verify if User is able to Fetch Single bookings		1.Login to POSTMAN tool 2.In POSTMan tool ,Import the Curl command using Raw text formal curl -i https://restful-booker.herokuapp.com/booking/x x-->x can be your preferred booking id 3.Click send request button and Verify the response body	1.Successful Login 2.Successful Navigation 3.The API must fetch the single booking which you requested				Vanessa
TC12				.VErify JSON Response for Get Single Booking Booking		1.Login to POSTMAN tool 2.In POSTMan tool ,Import the Curl command using Raw text formal curl -i https://restful-booker.herokuapp.com/booking/x x-->x can be your preferred booking id 3.Click send request button and Verify the response body	1.Successful Login 2.Successful Navigation 3.The Response body must show the booking requested				Vanessa

TC13				.Verify status code When all bookings are deleted		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text formal
						curl -i https://restful-booker.herokuapp.com/booking/x
						
						x-->x can be your preferred booking id
						
						3.Click send request button and Verify the response code
						Scripting:
						pm.test("Status code is 404", function () {
						pm.response.to.have.status(404);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 404
										Scripting:
										pm.test("Status code is 404", function () {
										pm.response.to.have.status(404);
										});
											Vanessa

TC14				.Verify Status code for Get Single booking		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text formal
						curl -i https://restful-booker.herokuapp.com/booking/x
						x-->x can be your preferred booking id
						3.Click send request button and Verify the response code
						Scripting:
						pm.test(""Status code is 200"", function () {
						pm.response.to.have.status(200);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 200
										Scripting:
										pm.test(""Status code is 200"", function ()
										pm.response.to.have.status(200);
										});
											Vanessa

											Vanessa

TC15	Create a New resourc	POST		Verify if user is able to Create New resource		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text format
						https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the response code
						Scripting:
						pm.test(""""Status code is 200"""", function () {
						pm.response.to.have.status(200);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 200
										pm.test(""""Status code is 200"""", functio
										pm.response.to.have.status(200);
										});
											Vanessa

TC16				Verify JSON Response for Successful POST Request		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text format
						https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the response
							1.Successful Login
							2.Successful Navigation
							3.The Response must be
							{
							"bookingid": 1,
							"booking": {
							"firstname": "Jim",
							"lastname": "Brown",
							"totalprice": 111,
							"depositpaid": true,
							"bookingdates": {
							"checkin": "2018-01-01",
							"checkout": "2019-01-01"
							},
							"additionalneeds": "Breakfast"
							}
							}
											Vanessa

TC17				Verify the status code for successful POST Request		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text format
						https://restful-booker.herokuapp.com/booking
						3.Click send request button and Verify the response code
						Scripting:
						pm.test(""""Status code is 200"""", function () {
						pm.response.to.have.status(200);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 200
										pm.test(""""Status code is 200"""", functio
										pm.response.to.have.status(200);
										});
											Vanessa

TC18				Verify if user is not able to create new resource using POST request when Request json body sent has only 4 out of 8 fields		1.Login to POSTMAN tool 2.In POSTMan tool ,Import the Curl command using Raw text format https://restful-booker.herokuapp.com/booking 3. In request body Enter only 4 out of 8 fields Click send request button and Verify the response code	Expected Result: 1.Successful Login 2.Successful Navigation 3.The Response code must be 500				Vanessa
TC19				Verify the status code of POST request when Request is sent with Request json body containing only 4 out of 8 fields		1.Login to POSTMAN tool 2.In POSTMan tool ,Import the Curl command using Raw text format https://restful-booker.herokuapp.com/booking 3. In request body Enter only 4 out of 8 fields Click send request button and Verify the response code	Expected Result: 1.Successful Login 2.Successful Navigation 3.The Response code must be 500				Vanessa
											Vanessa

TC20	Full update of a resource	PUT		Verify the Response body for Successful PUT request		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text format
						https://restful-booker.herokuapp.com/booking/:id
						id=tester preference
						
						curl command:
						curl -X PUT \
						https://restful-booker.herokuapp.com/booking/1 \
						-H 'Content-Type: application/json' \
						-H 'Accept: application/json' \
						-H 'Cookie: token=abc123' \
						-d '{
						"firstname" : "James",
						"lastname" : "Brown",
						"totalprice" : 111,
						"depositpaid" : true,
						"bookingdates" : {
						"checkin" : "2018-01-01",
						"checkout" : "2019-01-01"
						},
						"additionalneeds" : "Breakfast"
						}'
						3.Click send request button and Verify the response body and code
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 200 and response body must be
							HTTP/1.1 200 OK
							
							{
							"firstname" : "James",
							"lastname" : "Brown",
							"totalprice" : 111,
							"depositpaid" : true,
							"bookingdates" : {
							"checkin" : "2018-01-01",
							"checkout" : "2019-01-01"
							},
							"additionalneeds" : "Breakfast"
							}
											Vanessa

TC21	Full update of a resource			Verify the Response header for successful PUT request		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text format
						https://restful-booker.herokuapp.com/booking/:id
						id=tester preference
						
						curl -X PUT \
						https://restful-booker.herokuapp.com/booking/1 \
						-H 'Content-Type: application/json' \
						-H 'Accept: application/json' \
						-H 'Cookie: token=abc123' \
						-d '{
						"firstname" : "Virat",
						"lastname" : "Kohli",
						"totalprice" : 111,
						"depositpaid" : true,
						"bookingdates" : {
						"checkin" : "2018-01-01",
						"checkout" : "2019-01-01"
						},
						"additionalneeds" : "Breakfast"
						}'3.Click send request button and Verify the response header
						Scripting:
						pm.test("Content-Type is present", function () {
						pm.response.to.have.header("Content-Type");
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 200 and response header must be present
							Scripting:
							pm.test("Content-Type is present", function () {
							pm.response.to.have.header("Content-Type");
							});
										Scripting:
										pm.test("Content-Type is present", functio
										pm.response.to.have.header("Content-T
										});
											Vanessa

TC22	Full update of a resource			Verify the Status code for successful PUT request		1.Login to POSTMAN tool
						2.In POSTMan tool ,Import the Curl command using Raw text format
						https://restful-booker.herokuapp.com/booking/:id
						id=tester preference
						
						curl -X PUT \
						https://restful-booker.herokuapp.com/booking/1 \
						-H 'Content-Type: application/json' \
						-H 'Accept: application/json' \
						-H 'Cookie: token=abc123' \
						-d '{
						"firstname" : "Virat",
						"lastname" : "Kohli",
						"totalprice" : 111,
						"depositpaid" : true,
						"bookingdates" : {
						"checkin" : "2018-01-01",
						"checkout" : "2019-01-01"
						},
						"additionalneeds" : "Breakfast"
						}'3.Click send request button and Verify the status code
						Scripting:
						pm.test("Status code for successful PUT request is 200", function (
						{
						pm.response.to.have.status(200);
						});
							1.Successful Login
							2.Successful Navigation
							3.The Response code must be 200
										Scripting:
										pm.test("Status code for successful PUT r
										pm.response.to.have.status(200);
										});
											Vanessa

TC23	Full update of a resource			Verify if the user is not able to do partial update using PUT request		1.Login to POSTMAN tool
						2.Import the curl command using raw text format
						curl -X PUT \
						  https://restful-booker.herokuapp.com/booking/1 \
						  -H 'Content-Type: application/json' \
						  -H 'Accept: application/json' \
						  -H 'Cookie: token=abc123' \
						  -d '{
						    "firstname" : "James",
						    "lastname" : "Brown",
						    "totalprice" : 111,
						   // "depositpaid" : true,
						    "bookingdates" : {
						        "checkin" : "2018-01-01",
						        "checkout" : "2019-01-01"
						    },
						    "additionalneeds" : "Breakfast"
						}'
						
						3.Click send request button.Verify if user is not able to successfully update
						the booking
						
						POSTMAN Scripting:
						pm.test("Status code is 400", function () {
						    pm.response.to.have.status(400);
						});
						
						pm.test("Status code name has string", function () {
						    pm.response.to.have.status("Bad Request");
						});
							1.Sucessful login
							2.Import request successful
							3.The user must not be able to update the resource bookingid
							and response statuscode must be 400 bad request
										pm.test("Status code is 400", function () {
										    pm.response.to.have.status(400);
										});
										
										pm.test("Status code name has string", fur
										    pm.response.to.have.status("Bad Reque
										});
											Vanessa

TC24	Full update of a resou			Verify the status code that user gets for trying Partial update using PUT request		1.Login to POSTMAN tool
						2.Import the curl command using raw text format
						curl -X PUT \
						  https://restful-booker.herokuapp.com/booking/1 \
						  -H 'Content-Type: application/json' \
						  -H 'Accept: application/json' \
						  -H 'Cookie: token=abc123' \
						  -d '{
						    "firstname" : "James",
						    "lastname" : "Brown",
						    "totalprice" : 111,
						   // "depositpaid" : true,
						    "bookingdates" : {
						        "checkin" : "2018-01-01",
						        "checkout" : "2019-01-01"
						    },
						    "additionalneeds" : "Breakfast"
						}'
						
						3.Click send request button.Verify if user is not able to successfully update
						the booking
						
						POSTMAN Scripting:
						pm.test("Status code is 400", function () {
						    pm.response.to.have.status(400);
						});
						
						pm.test("Status code name has string", function () {
						    pm.response.to.have.status("Bad Request");
						});
							1.Sucessful login
							2.Import request successful
							3.The user must not be able to update the resource bookingid
							and response statuscode must be 400 bad request
										pm.test("Status code is 400", function () {
										    pm.response.to.have.status(400);
										});
										
										pm.test("Status code name has string", fur
										    pm.response.to.have.status("Bad Reque
										});
											Vanessa

											Vanessa

TC25	Partial Update			Verify the JSON Response body for PATCH request		1.Login to POSTMAN tool
						2.Import the curl command using raw text format
						curl -X PUT \
						  https://restful-booker.herokuapp.com/booking/938 \
						  -H 'Content-Type: application/json' \
						  -H 'Accept: application/json' \
						  -H 'Cookie: token=abc123' \
						  -d '{
						    "firstname" : "QATesterFirstnam",
						    "lastname" : "QAtesterLastName"
						}'
						3.Veirfy if user is be able to partially update resource booking 938 successfully
						4.verify the response body of patch request
							1.Sucessful login
							2.Import request successful
							3.The user must be able to do Patch update successfully and response status code must be 200 OK
							4.The response body must be as below
							
							{
							    "firstname": "QATesterFirstName",
							    "lastname": "QATesterLastName",
							    "totalprice": 0,
							    "depositpaid": false,
							    "bookingdates": {
							        "checkin": "2018-01-07",
							        "checkout": "2019-01-07"
							    },
							    "additionalneeds": "Breakfast,Lunch,Dinner"
							}
											Vanessa

TC26	Partial Update			Verify the Response header for PATCH Request							Vanessa

TC27	Partial Update	PATCH		Verify the status code for successful PATCH Request		1.Login to POSTMAN tool
						2.Import the curl command using raw text format
						curl -X PUT \\
						  https://restful-booker.herokuapp.com/booking/938 \\
						  -H 'Content-Type: application/json' \\
						  -H 'Accept: application/json' \\
						  -H 'Cookie: token=abc123' \\
						  -d '{
						    "firstname" : "QATesterFirstnam",
						    "lastname" : "QAtesterLastName"
						}'
						3.Verify if user is be able to partially update resource booking 938 successfully
						4.verify the status code for successful patch request
						
						POSTMAN SCRIPT:
						pm.test("Successful PATCH Request", function () {
						    pm.expect(pm.response.code).to.be.oneOf([200, 200]);
						});
							1.Sucessful login
							2.Import request successful
							3.The user must be able to do Patch update successfully and response status code must be 200 OK
							4.The response body must be as below
							
							{
							    ""firstname"": ""QATesterFirstName"",
							    ""lastname"": ""QATesterLastName"",
							    ""totalprice"": 0,
							    ""depositpaid"": false,
							    ""bookingdates"": {
							        ""checkin"": ""2018-01-07"",
							        ""checkout"": ""2019-01-07""
							    },
							    ""additionalneeds"": ""Breakfast,Lunch, Dinner""
							}
										pm.test("Successful PATCH Request", function () {
										    pm.expect(pm.response.code).to.be. oneOf([200, 200]);
										});
											Vanessa

TC28	Partial Update			Verify if user is able to perform PATCH request successfully without authorization		1.Login to POSTMAN tool 2.Import the curl command using raw text format curl -X PUT \\ https://restful-booker.herokuapp.com/booking/938 \\ -H 'Content-Type: application/json' \\ -H 'Accept: application/json' \\ -H 'Cookie: token=abc123' \\ -d '{ "firstname" : "QATesterFirstnam", "lastname" : "QAtesterLastName" }' 3.Navigate to authorization tab and make sure a)you dont provide credentials to authorize the Patch request b)Leave it as blank 4.Click send request button 5.Verify if user is not able to perform PATCH request	1.Sucessful login 2.Import request successful 3.No authorization provided for the patch request 4.Successful action 5.The patch request must fail and status code must be 403				Vanessa
TC29	Partial Update			Verify if user is able to perform PATCH request by providing JSON request body as blank							Vanessa

TC30	Partial Update			Verify is user is getting correct error code for failed PATCH request		1.Login to POSTMAN tool
						2.Import the curl command using raw text format
						curl -X PUT \\
						  https://restful-booker.herokuapp.com/booking/938 \\
						  -H 'Content-Type: application/json' \\
						  -H 'Accept: application/json' \\
						  -H 'Cookie: token=abc123' \\
						  -d '{
						    ""firstname"" : ""QATesterFirstnam"",
						    ""lastname"" : ""QAtesterLastName""
						}'
						3.Navigate to authorization tab and make sure
						a)you dont provide credentials to authorize the Patch request
						b)Leave it as blank
						4.Click send request button
						5.Verify is user is getting correct error code for failed PATCH request
						6.Provide authorization and change the input field name against the requirement and send the patch request
						{
						    "Helloname" : "QA",
						    "middlename" : "QAlast"
						}
						7.Verify the status code
							1.Sucessful login
							2.Import request successful
							3.No authorization provided for the patch request
							4.Successful action
							5.The patch request must fail and status code must be 403
							6.Patch request must fail
							7.Status code must be 400 for incorrect syntax
											Vanessa

TC31	Delete			.Verify if user is able to delete all bookings		1.Login to POSTMAN tool 2.Import the curl command using raw text format curl -X DELETE \\ https://restful-booker.herokuapp.com/booking/2952 \\ -H 'Content-Type: application/json' \\ -H 'Cookie: token=abc123' 3.Click send request 4.Verify if user is able to delete the booking successfully 5.Verify the status code	1.Successful login 2.Import request successful 3.User must be able to perform SEND rrequest 4.The user must be able delete booking id 2952 successfully if it is not previously deleted 5.The status code must be 200				Vanessa