Capstone Project	Student Name: Vijina P			
API Testing for Contact List Application				

Github repository

The source code and this document has been uploaded to the following git repository;

Url	https://github.com/vijinabalakrishnan/Capstone_Projects/tree/main/API_Testing_Contact_List_Application
Branc h	main
Acces	Public
S	

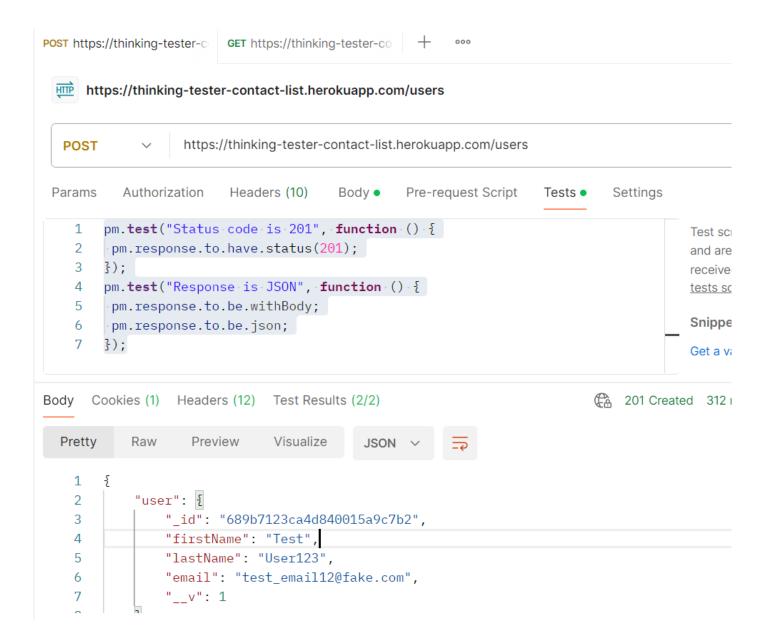
```
Test Case 1: Add New User
```

```
Request Type: POST
Endpoint: <a href="https://thinking-tester-contact-list.herokuapp.com/users">https://thinking-tester-contact-list.herokuapp.com/users</a>
Request Payload:
{
        "firstName": "Test",
        "lastName": "User123",
        "email": "test_email12@fake.com",
        "password": "myPassword"
}
Postman Chai Assertions
pm.test("Status code is 201", function () {
pm.response.to.have.status(201);
});
pm.test("Response is JSON", function () {
pm.response.to.be.withBody;
pm.response.to.be.json;
});
Response
{
  "user": {
    "_id": "689b7123ca4d840015a9c7b2",
    "firstName": "Test",
    "lastName": "User123",
    "email": "test_email12@fake.com",
    " v": 1
  },
  "token": "eyJhbGciOiJIUzI1NilsInR5cCl6lkpXVCJ9.eyJfaWQiOiI2ODliNzEyM2NhNGQ4NDAwMTVhOWM3YjliLCJpYXQ
```

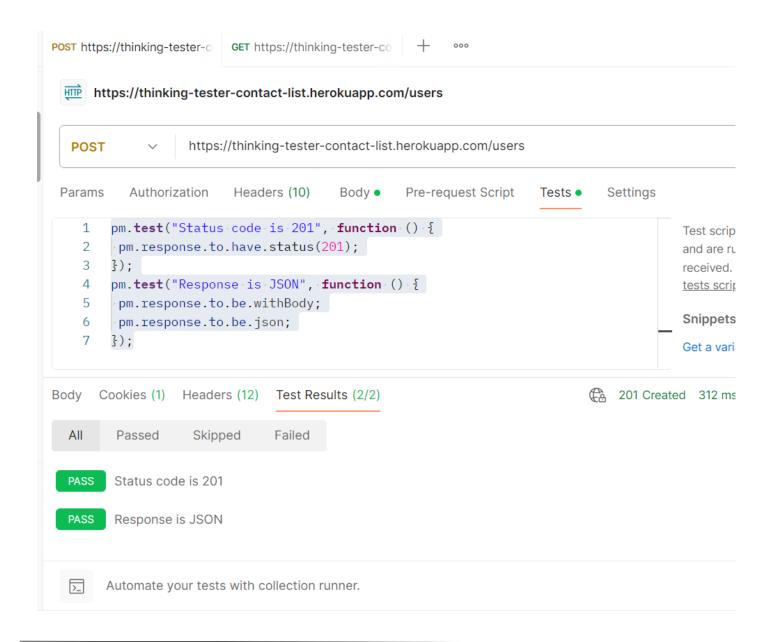
Response Screenshot

}

iOjE3NTUwMTc1MDd9.QS68bNqQpuc7Q0f1g4lc_LDFigtpPBTx0cBcgJYr6hc"



Test Results Screenshot



Test Case 2:

Get user Profile Request Type: GET

Endpoint: https://thinking-tester-contact-list.herokuapp.com/users/me

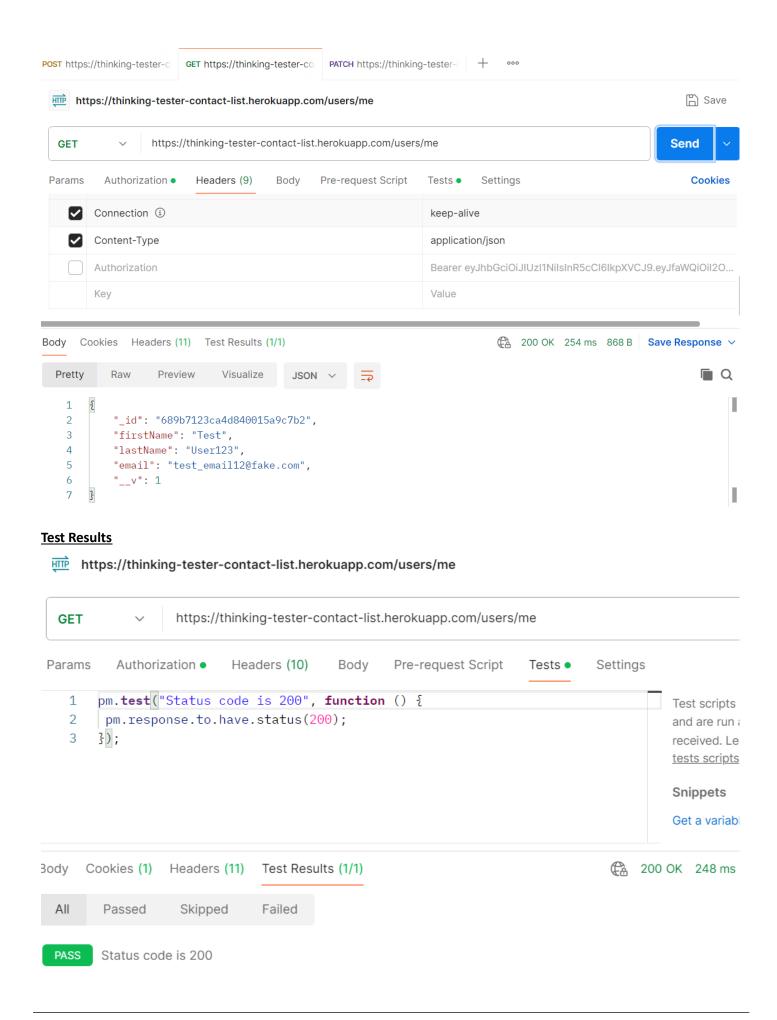
Headers: Authorization Bearer {{token}}

Response Validation: Test for status code and message 200 OK

Postman Chai Assertions

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

Response screenshot



```
Request Type: PATCH
Endpoint: <a href="https://thinking-tester-contact-list.herokuapp.com/users/me">https://thinking-tester-contact-list.herokuapp.com/users/me</a>
Headers: Authorization Bearer {{token}}
Request Payload:
{
         "firstN ame": "Updated",
         "lastName": "Username",
         "email": "test_email12@fake.com",
         "password": "myNewPassword"
}
Validate for status code and message 200 OK
Postman Chai Assertions
pm.test("Status code is 200", function () {
pm.response.to.have.status(200);
});
Response:
 POST https://thinking-teste × GET https://thinking-tester-co
                                                          PATCH https://thinking-tester-
   https://thinking-tester-contact-list.herokuapp.com/users/me
                                                                                                                                 Sa<sup>1</sup>
    PATCH
                       https://thinking-tester-contact-list.herokuapp.com/users/me
                                                                                                                              Send
  Params
             Authorization •
                               Headers (11)
                                                Body •
                                                          Pre-request Script
                                                                               Tests •
                                                                                          Settings
                                                                                                                                  Cooki
            Hide auto-generated headers
  Headers
           Key
                                                                             Value
                                                                                                                                    Bu
           Authorization (1)
                                                                             Bearer eyJhbGciOiJIUzl1NilsInR5cCl6lkpXVCJ9.eyJfaWQiOil2
           Postman-Token (1)
                                                                              <calculated when request is sent>
           Content-Type (i)
                                                                             text/plain
 Body
        Cookies Headers (11)
                                 Test Results (1/1)
                                                                                           200 OK 1324 ms 861 B
                                                                                                                         Save Response
    Pretty
              Raw
                        Preview
                                     Visualize
                                                   JSON
                                                                                                                                   1
```

Test Result

2

3

4 5

6

"_id": "689b7123ca4d840015a9c7b2",

"email": "test_email12@fake.com",

"firstName": "Test",

"__v": 1

"lastName": "Username",





Test Case 4:

Log In User

Request Type: POST

Endpoint: https://thinking-tester-contact-list.herokuapp.com/users/login

Request Payload:

{ "email": "test_email12@fake.com", "password": "myNewPassword" }

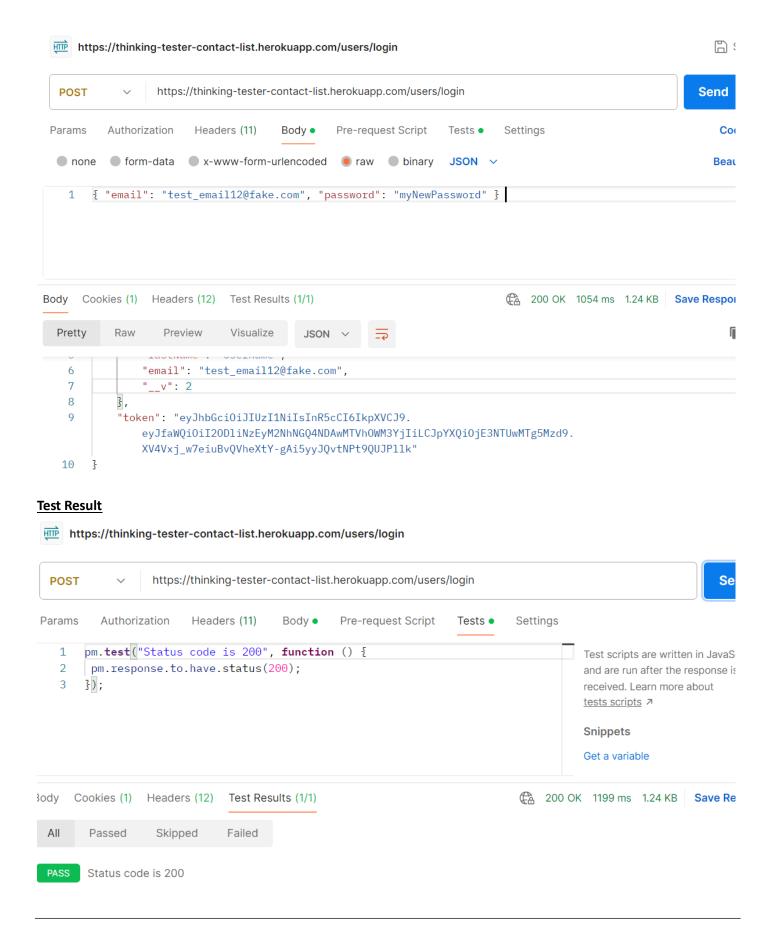
Test it in Postman and Get the Token and validate for status code 200 OK

Postman Chai Assertions

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

Response:
{
    "user": {
        "_id": "689b7123ca4d840015a9c7b2",
        "firstName": "Test",
        "lastName": "Username",
        "email": "test_email12@fake.com",
        "__v": 2
    },
    "token": "eyJhbGciOiJIUzl1NilsInR5cCl6lkpXVCJ9.eyJfaWQiOiI2ODliNzEyM2NhNGQ4NDAwMTVhOWM3YjliLCJpYXQ iOjE3NTUwMTg5Mzd9.XV4Vxj_w7eiuBvQVheXtY-gAi5yyJQvtNPt9QUJPllk"
}
```

Screenshot



Test Case 5:

Add Contact

Request Type: POST

Endpoint: https://thinking-tester-contact-list.herokuapp.com/contacts

Headers: Authorization Bearer {{token}}

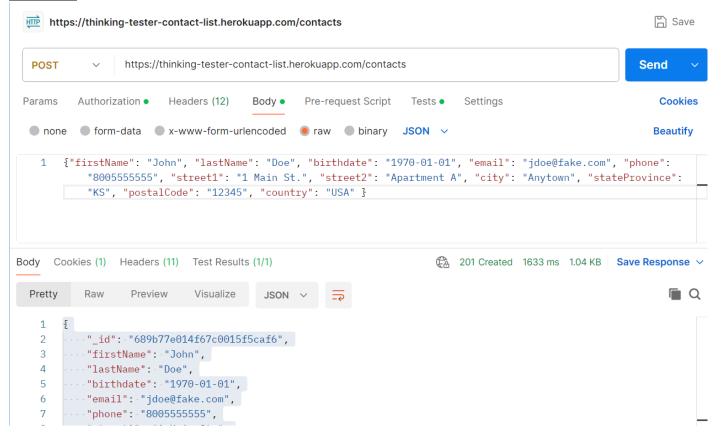
```
Request Payload: { "firstName": "John", "lastName": "Doe", "birthdate": "1970-01-01", "email": "jdoe@fake.com", "phone": "8005555555", "street1": "1 Main St.", "street2": "Apartment A", "city": "Anytown", "stateProvince": "KS", "postalCode": "12345", "country": "USA" }
```

Validate for 201 Created

```
Postman Chai Assertions
```

```
pm.test("Status code is 201", function () {
pm.response.to.have.status(201);
});
Response:
  "_id": "689b77e014f67c0015f5caf6",
  "firstName": "John",
  "lastName": "Doe",
  "birthdate": "1970-01-01",
  "email": "jdoe@fake.com",
  "phone": "8005555555",
  "street1": "1 Main St.",
  "street2": "Apartment A",
  "city": "Anytown",
  "stateProvince": "KS",
  "postalCode": "12345",
  "country": "USA",
  "owner": "689b7123ca4d840015a9c7b2",
  " v": 0
}
```

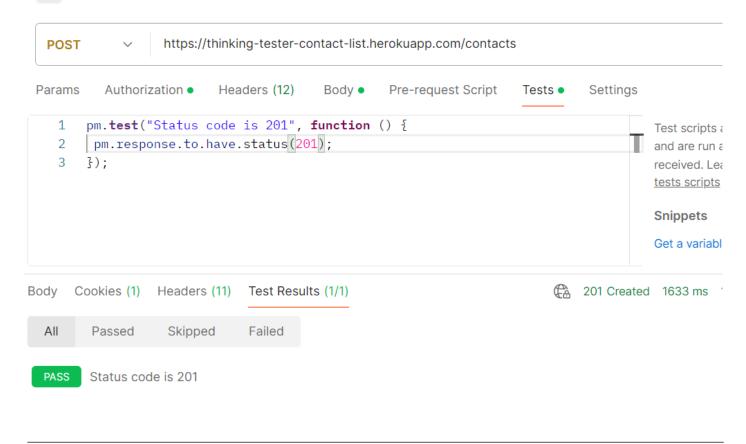
Screenshot



Test Results



https://thinking-tester-contact-list.herokuapp.com/contacts



Test Case 6:

Get Contact List Request Type: GET

Endpoint: https://thinking-tester-contact-list.herokuapp.com/contacts

Headers: Authorization Bearer {{token}}

Validate for 200 OK

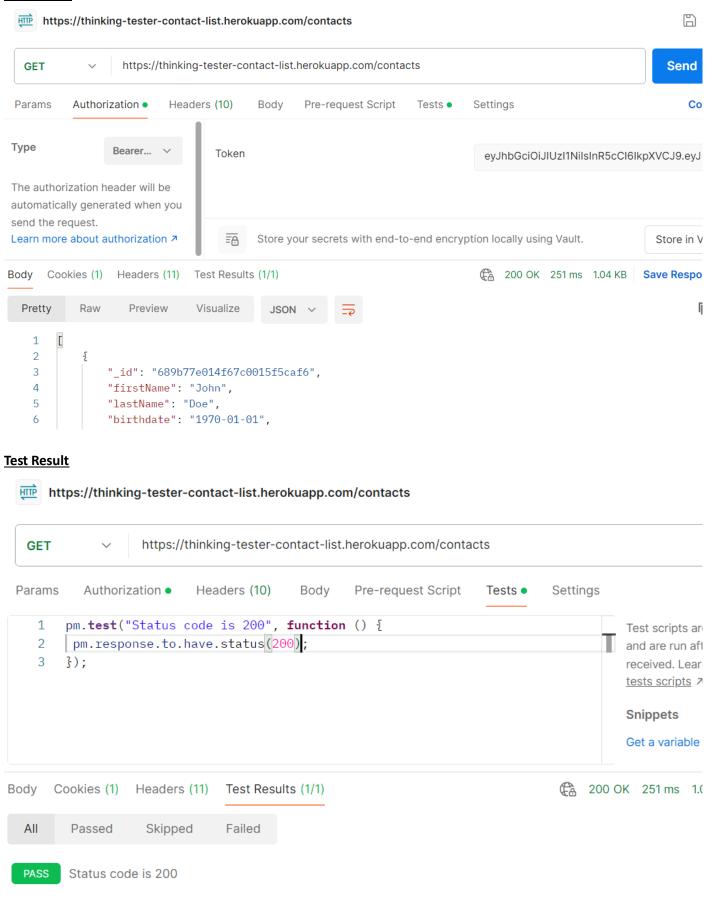
Postman Chai Assertions

"postalCode": "12345",

```
pm.test("Status code is 200", function () {
pm.response.to.have.status(200);
});
Response
    "_id": "689b77e014f67c0015f5caf6",
    "firstName": "John",
    "lastName": "Doe",
    "birthdate": "1970-01-01",
    "email": "jdoe@fake.com",
    "phone": "8005555555",
    "street1": "1 Main St.",
    "street2": "Apartment A",
    "city": "Anytown",
    "stateProvince": "KS",
```

```
"country": "USA",
"owner": "689b7123ca4d840015a9c7b2",
"__v": 0
}
```

Screenshot



```
Test Case 7:
```

Get Contact

Request Type: GET

Endpoint: https://thinking-tester-contact-list.herokuapp.com/contacts/689b77e014f67c0015f5caf6

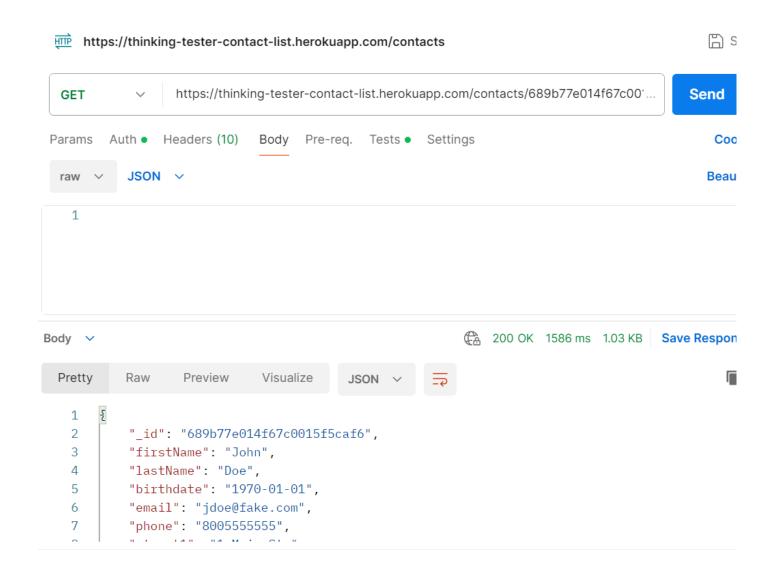
Headers Authorization Bearer {{token}}

Validate for 200 OK

Postman Chai Assertions

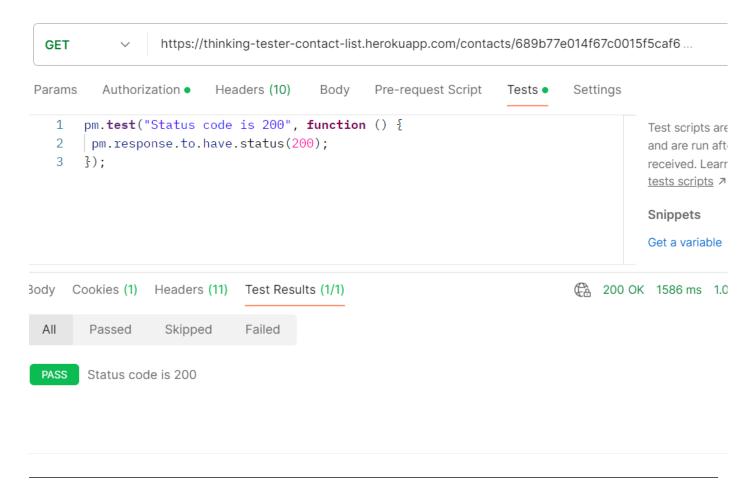
```
pm.test("Status code is 200", function () {
pm.response.to.have.status(200);
});
Response
{
         "_id": "689b77e014f67c0015f5caf6",
         "firstName": "John",
         "lastName": "Doe",
         "birthdate": "1970-01-01",
         "email": "jdoe@fake.com",
         "phone": "800555555",
         "street1": "1 Main St.",
         "street2": "Apartment A",
         "city": "Anytown",
         "stateProvince": "KS",
         "postalCode": "12345",
         "country": "USA",
         "owner": "689b7123ca4d840015a9c7b2",
         " v": 0
}
```

SCreenshot



Test Result

https://thinking-tester-contact-list.herokuapp.com/contacts



Test Case 8:

Update Contact

```
Request Type: PUT
```

End Point: https://thinking-tester-contact-list.herokuapp.com/contacts/689b77e014f67c0015f5caf6

```
Headers Authorization Bearer {{token}}
```

```
Request Payload

{

"firstName": "Amy",

"lastName": "Miller",

"birthdate": "1992-02-02",

"email": "amiller@fake.com",

"phone": "8005554242",

"street1": "13 School St.",

"street2": "Apt. 5",

"city": "Washington",

"stateProvince": "QC",

"postalCode": "A1A1A1",

"country": "Canada"
```

Validate for Email and 200 OK

Postman Chai Assertions

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

Response Screenshot

3ody ∨

3

PUT

https://thinking-tester-contact-list.herokuapp.com/contacts/689b77e014f67c0015

Ser

Params Auth

Headers (12) Body

Pre-req. Tests

Settings

https://thinking-tester-contact-list.herokuapp.com/contacts/689b77e014f67c0015f5caf6

```
1 {
2    "firstName": "Amy",
3    "lastName": "Miller",
4    "birthdate": "1992-02-02",
5    "email": "amiller@fake.com",
6    "phone": "8005554242",
```

200 OK 315 ms 1.04 KB

Save Res

"firstName": "Amv".

Test Result

https://thinking-tester-contact-list.herokuapp.com/contacts/689b77e014f67c0015f5caf6



Test Case 9:

Update Contact Request Type: PATCH

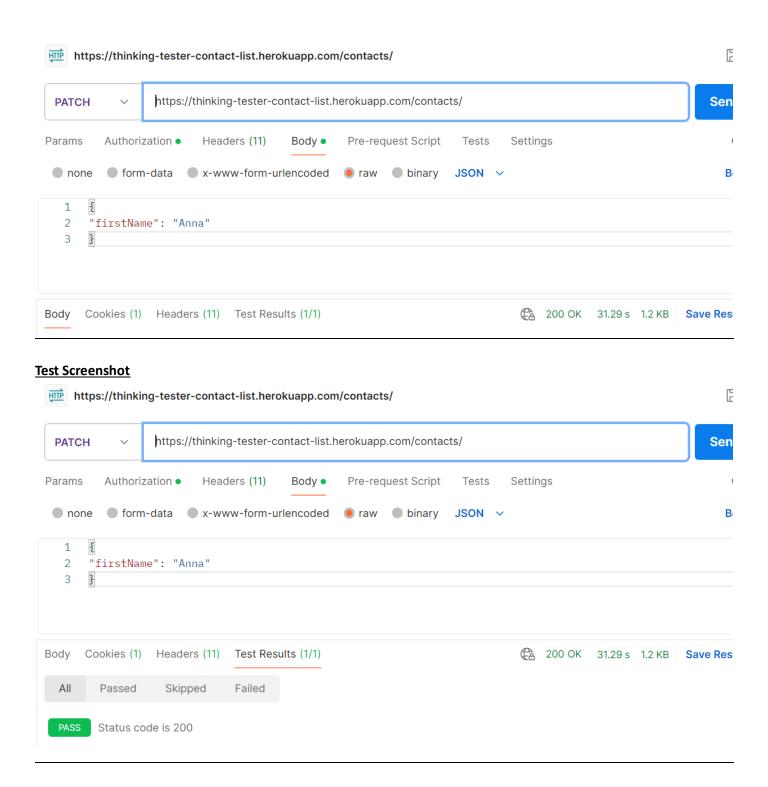
Endpoint: https://thinking-tester-contact-list.herokuapp.com/contacts/

Headers Authorization Bearer {{token}}
Request Payload: { "firstName": "Anna" }
Validate for first name and 200 OK

Postman Chai Assertions

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

Response screenshot



Test Case 10:

Logout User

Request Type: POST

Endpoint: https://thinking-tester-contact-list.herokuapp.com/users/logout

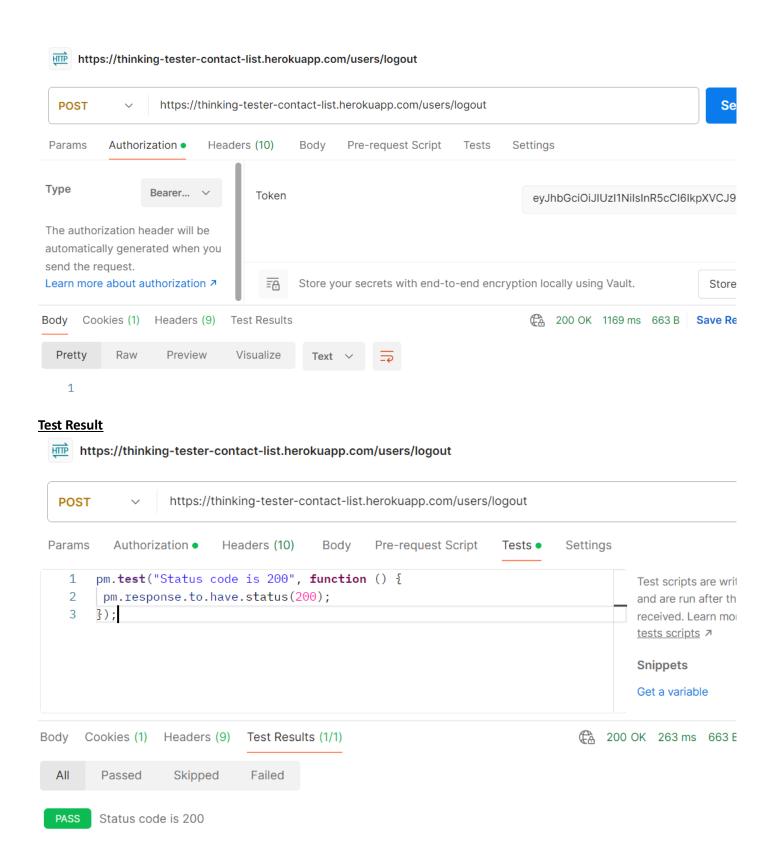
Headers Authorization Bearer {{token}}

Validate for 200 OK

Postman Chai Assertions

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

Response screenshot



RestAssured test cases

Checked into the below github repository(Same as above)

Url	https://github.com/vijinabalakrishnan/Capstone Projects/tree/main/API_Testing_Contact_List_Appli
	cation
Branc	main
h	

Acces	Public
S	

Source code:

/src/test/java/RestAPITesting.java

```
package test.java;
import io.restassured.RestAssured;
import io.restassured.http.ContentType;
import io.restassured.response.Response;
import org.testng.Assert;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;
public class RestAPITesting {
        private String token;
        private String contactId;
  //@Test --Uncomment to register a new user
  public void testAddNewUser() {
    RestAssured.baseURI = "https://thinking-tester-contact-list.herokuapp.com";
    String requestBody = """
          "firstName": "Test",
                                       "lastName": "User123",
                                       "email": "test_email125@fake.com",
                                       "password": "myPassword"
        nnn,
    Response response = given()
        .contentType(ContentType.JSON)
        .body(requestBody)
    .when()
        .post("/users")
    .then()
        .statusCode(201)
        .statusLine(equalTo("HTTP/1.1 201 Created"))
        .extract().response();
    String token = response.jsonPath().getString("token");
    token.codePoints();
```

```
public void setupAndLogin() {
  RestAssured.baseURI = "https://thinking-tester-contact-list.herokuapp.com";
  // Login to get token (assumes user already exists)
  String loginBody = """
        "email": "test_email125@fake.com",
        "password": "myPassword"
      nnn,
  Response loginResponse = given()
      .contentType(ContentType.JSON)
      .body(loginBody)
  .when()
      .post("/users/login")
  .then()
      .statusCode(200)
      .extract().response();
  token = loginResponse.jsonPath().getString("token");
@Test(priority = 1)
public void testGetUserProfile() {
  given()
      .header("Authorization", "Bearer " + token)
  .when()
      .get("/users/me")
  .then()
      .statusCode(200)
      .statusLine(equalTo("HTTP/1.1 200 OK"));
}
@Test(priority = 2)
public void testUpdateUser() {
  String updateBody = """
        "firstName": "Updated",
        "lastName": "Username",
        "email": "test email125@fake.com",
                                     "password": "myPassword"
      000g
  given()
      .header("Authorization", "Bearer " + token)
      .contentType(ContentType.JSON)
      .body(updateBody)
  .when()
      .patch("/users/me")
  .then()
      .statusCode(200)
```

```
.statusLine(equalTo("HTTP/1.1 200 OK"));
@Test(priority = 3)
public void testAddContact() {
  String contactBody = """
         "firstName": "John",
        "lastName": "Doe",
         "birthdate": "1970-01-01",
         "email": "test email125@fake.com",
         "phone": "800555555",
         "street1": "1 Main St.",
        "street2": "Apartment A",
        "city": "Anytown",
        "stateProvince": "KS",
        "postalCode": "12345",
        "country": "USA"
      """;
  given()
      .header("Authorization", "Bearer " + token)
      .contentType(ContentType.JSON)
      .body(contactBody)
  .when()
      .post("/contacts")
  .then()
      .statusCode(201)
      .statusLine(equalTo("HTTP/1.1 201 Created"));
@Test(priority = 4)
public void testGetContactList() {
     Response searchResponse = given()
      .header("Authorization", "Bearer " + token)
      .contentType(ContentType.JSON)
  .when()
      .get("/contacts")
  .then()
      .statusCode(200)
      .statusLine(equalTo("HTTP/1.1 200 OK"))
      .extract().response();
     contactId = searchResponse.jsonPath().getString("[0]._id");//store the search result
}
@Test(priority = 5)
public void testGetContactById() {
  given()
      .header("Authorization", "Bearer " + token)
      .contentType(ContentType.JSON)
  .when()
```

```
.get("/contacts/" + contactId)
  .then()
      .statusCode(200)
      .statusLine(equalTo("HTTP/1.1 200 OK"));
@Test(priority = 6)
public void testUpdateContact() {
     System.out.println("First Contact ID: " + contactId);
  String updateBody = """
      {
                                      "firstName": "Amy",
                                     "lastName": "Miller",
                                     "birthdate": "1992-02-02",
                                     "email": "test_email125@fake.com",
                                     "phone": "8005554242",
                                      "street1": "13 School St.",
                                     "street2": "Apt. 5",
                                     "city": "Washington",
                                      "stateProvince": "QC",
                                     "postalCode": "A1A1A1",
                                      "country": "Canada"
      nnn;
  given()
      .header("Authorization", "Bearer " + token)
      .contentType(ContentType.JSON)
      .body(updateBody)
  .when()
      .put("/contacts/" + contactId)
  .then()
      .statusCode(200)
      .body("email", equalTo("test_email125@fake.com"))
      .statusLine(equalTo("HTTP/1.1 200 OK"));
}
@Test(priority = 7)
public void testPatchUpdateContact() {
  String patchBody = """
        "firstName": "Anna"
      """;
  given()
      .header("Authorization", "Bearer " + token)
      .contentType(ContentType.JSON)
      .body(patchBody)
  .when()
      .patch("/contacts/" + contactId)
  .then()
      .statusCode(200)
```