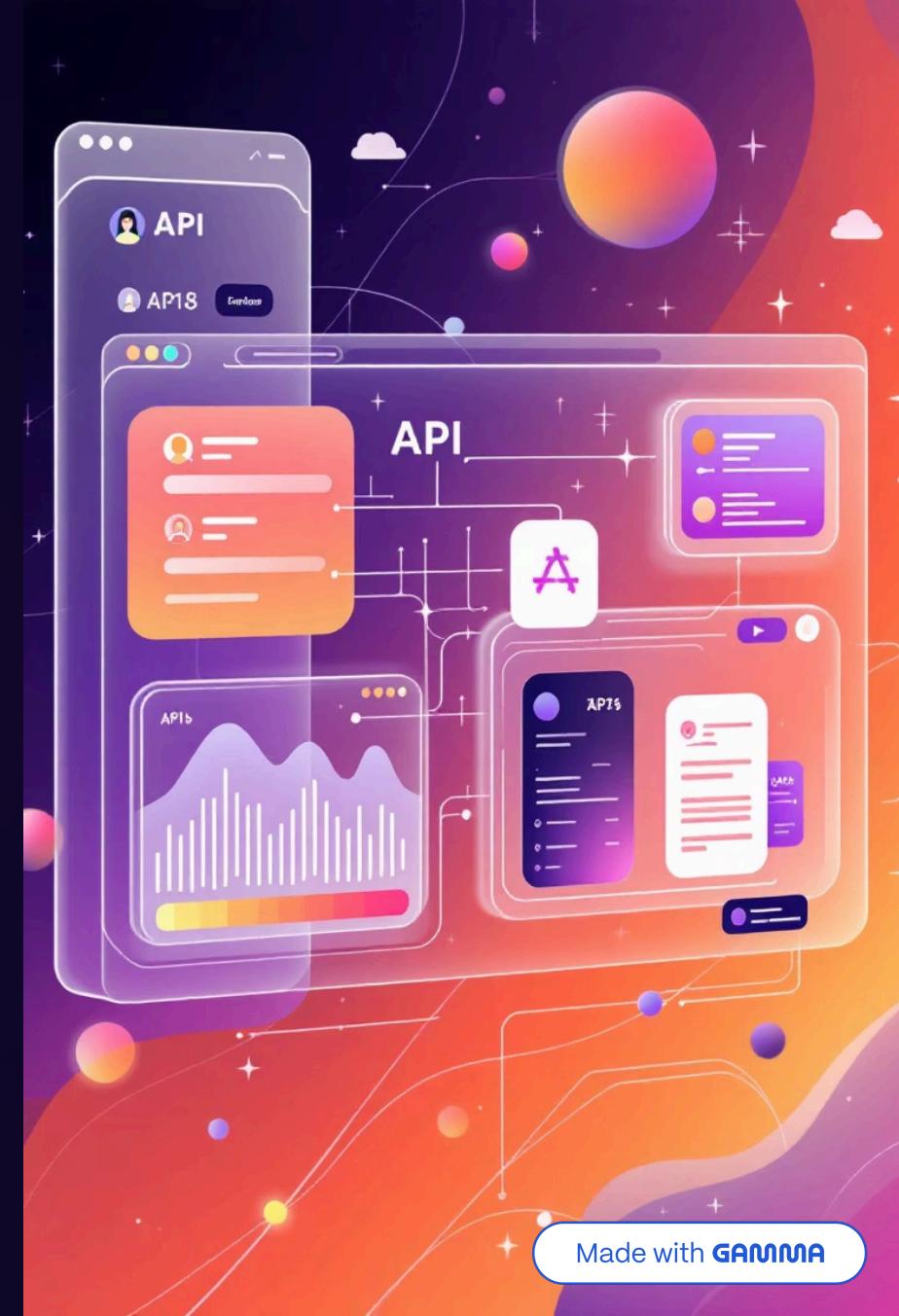


# API Testing Project: Contact List API

A comprehensive API testing project showcasing robust testing methodologies using Postman and Newman.





# Project Overview & Objective



## Objective

Validate the functionality and reliability of the Contact List API through automated testing.



## API Under Test

Contact List API: Enables user and contact management (add, edit, delete).



## Tools Used

Postman for API development and testing, Newman for CLI execution.



## Authentication Flow: JWT

1

### User Login

POST request with valid credentials  
(email, password).

2

### JWT Generation

API generates a JSON Web Token  
upon successful authentication.

3

### Token Usage

JWT used as a Bearer Token for all  
subsequent API requests.

# CRUD API Test Coverage

Comprehensive testing of core API functionalities for contact management.

## GET Requests

Retrieve all contacts and specific contacts by ID.

## POST Request

Add new contact records to the list.

## PUT Request

Update existing contact details using contact ID.

## DELETE Request

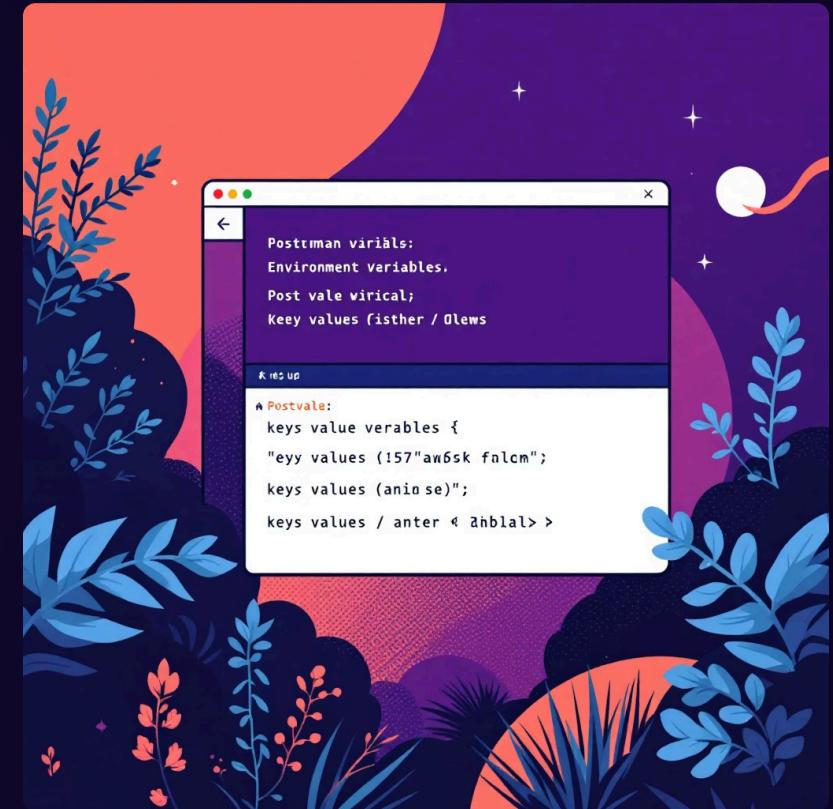
Remove specific contact records by ID.



# Environment Variables for Dynamic Testing

Leveraged Postman's environment variables to manage dynamic data and authentication tokens.

- **Initial Value:** Stored in the environment file.
- **Current Value:** Temporary, used during execution.
- Programmatically saved response data (e.g., Contact ID, JWT) for reuse across requests.





# Positive Test Scenarios

Ensuring expected functionality under normal operating conditions.

## Successful Login

Validates user authentication and JWT generation.

## Contact Creation

Confirms new contacts are added correctly.

## Data Retrieval

Verifies accurate fetching of all and specific contact details.

## Contact Update & Deletion

Ensures modifications and removals are processed as expected.



# Negative Test Scenarios

Validating API behavior under erroneous or unexpected conditions.

## Missing Authentication

Request sent without a valid token.

## Record Not Found

Attempting to access a non-existent contact ID.

## Missing Required Info

POST/PUT requests with incomplete data (e.g., no first name).

## Invalid Data Format

Submitting data with incorrect types or lengths (e.g., invalid email, overlong last name).

# Assertions and Validations Strategy

Ensuring API responses meet predefined criteria.

1

## Status Code Assertions

Verifying HTTP status codes (e.g., 200 OK, 201 Created, 401 Unauthorized, 404 Not Found).

2

## Body Assertions

Validating the content of the response body, including specific strings and JSON structure.

3

## Header & Response Time

Checking response headers and ensuring timely API responses.

4

## Nested JSON Assertions

Deep validation of complex JSON structures and nested field values.



```
<assertion onager((asster) (cavitation("range//epbe"))),>
  ciationEvaluation ()>
    <unification-un(l),>
      apvien("-assuson()")>
        {(-farlary);
         <elvigerous('ansulation'))},))>
          <erivation(antivomn')>l )>

  assertor::campan(ervclente
    dirch'vlentianntion
      tarivacts )>
    <perion ann(ric
      <aristis an/c
        (linicallo
        (cavulation""an
        <tel);
        falvisation(
          lmt (:ather
            <perivien(
              ))>

    sit <clatom>
    <canization vation t
      <itter v(cande)+>
      <gdrivaliadmaty>
        <clatten>
        <leicat(/ambey-action.
          ((.carditionistp))>
        <itter/fiction <();
        <remoration(lentylow/anssterictiom ca/ (lessunt(-(*)>
          <alilidation /cateriom= (assutem">
            dstering action');
        >
      <trvidation (canubers)>
        <driven'(canour(lasster)->
          vintier">
            appication (hen))))je;
      >>
    <
      <allvalication cano clore );
      <rrvalicates> suttom((liom));
        <lingschamazion viaster), (ltan conanger )>
          <t-ravulation cutictes-lapt
```

# CLI Execution with Newman

Automated execution of Postman collections via the command line.

- **Seamless Integration:** Newman allows running collections without the Postman GUI.
- **Export & Run:** Collections and environment variables are exported and executed via CLI.
- **CI/CD Readiness:** Facilitates integration into continuous integration/continuous deployment pipelines.



# Business Value & Skills Demonstrated



## Improved Quality

Ensured API reliability and functionality, reducing potential bugs in production.



## Efficiency Gains

Automated testing reduces manual effort and accelerates release cycles.



## Enhanced Security

Validated authentication flows and error handling for robust security.



## Technical Proficiency

Demonstrated expertise in Postman, Newman, API testing, and automation.

