Complete `README.md` for setting up and running the project

By following this guide, you should be able to set up, run, and verify the API automation project. If you face any issues, feel free to raise them in the repository or contact me(Abdullah).

Overview

This project automates the validation of users from the city "FanCode," ensuring that more than 50% of their `todo` tasks are completed. The automation interacts with the public APIs from `JSONPlaceholder`.

The following criteria are checked:-
- Users are identified as belonging to the "FanCode city" if their latitude is between `-40` and `5` and their longitude is between `5` and `100`.
- For each user in "FanCode city," the percentage of completed tasks is calculated.
- The percentage of completed tasks should be greater than 50% for the user to pass the validation.

Technologies Used:-
- Java: Main programming language for automation.
- RestAssured: For interacting with REST APIs.
- TestNG: Used for writing and organizing test cases.
- Maven: For project build, dependency management, and test execution.
- Gson: For parsing JSON responses into Java objects.
- JSONPlaceholder APIs: Mock API for users and todos.

Project Structure:-

```
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── models
│   │   │   ├── services
│   │   └── resources
│   └── test
│       ├── java
│       │   └── tests
```

```
├── pom.xml
└── README.md
```

- **models**: Contains `User` and `Todo` classes that represent the data models.
- **services**: Contains classes for interacting with the `users` and `todos` APIs.
- **tests**: Contains the test cases for validating the user task completion.

Prerequisites:-

Ensure the following tools are installed on your system before setting up the project:

- Java Development Kit (JDK): Java 8 or higher.
- Maven: For building the project and managing dependencies.
- Git: For cloning the repository.
- An IDE (like IntelliJ IDEA or Eclipse): To open and edit the project.

Installation and Setup

Step 1: Clone the Repository

```
git clone https://github.com/your-username/fancode-api-automation.git
cd fancode-api-automation
```

Step 2: Build the Project

Use Maven to build the project and download necessary dependencies:

```
mvn clean install
```

Step 3: Run the Tests

Run the tests using Maven, which will execute the TestNG test suite:

```
mvn test
```

Step 4: Verify the Output

Upon running the tests, the result will display whether users from FanCode city have more than 50% of their tasks completed. If all users pass the criteria, the output will look like this:

```
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
```

If any user fails (i.e., they have less than 50% task completion), you will see a failure message like this:
java. lang.AssertionError: User Leanne Graham has less than 50% task completion.

Project Details

APIs Used:-

- Users API: `https://jsonplaceholder.typicode.com/users`
- Todos API: `https://jsonplaceholder.typicode.com/todos`

User Filtering Criteria:-

A user belongs to "FanCode city" if:

- Latitude (`lat`) is between `-40` and `5`.
- Longitude (`lng`) is between `5` and `100`.

Task Validation Criteria:-

For each user in "FanCode city":

- The completed task percentage is calculated based on the total number of tasks.
- The test asserts that the completed task percentage is greater than 50%.

Troubleshooting:-

- Maven build issues: Ensure that Maven is installed and properly configured in your system's environment variables.
- Java version compatibility: Use Java 8 or higher. You can check your version by running `java -version` in your terminal.
- Failed tests: Verify the JSONPlaceholder API is accessible and make sure the test criteria align with the data in the API.