# **IT314: Software Engineering**

**Unit testing of backend API**

****

**Group: 15 Conference Management System**

|  |  |
| --- | --- |
| **IDs** | **Member Name** |
| 202001117 | PARVA BHARATKUMAR DHAMI |
| 202001122 | MONPARA SUMIT RAJESHBHAI |
| 202001128 | MEGHABEN RATHWA |
| 202001132 | SONAGARA SARTHAK MOHANBHAI |
| 202001154 | METKEL HARSHKUMAR ANANDBHAI |
| 202001162 | PADHIYAR KARAN HARESHBHAI |
| 202001172 | PATEL VEDANT HARSHADBHAI |
| 202001176 | MANGUKIYA JENISH MAHESHBHAI |
| 202001177 | PRAJAPATI PARTH SURESHBHAI |

**Overview**

Unit testing is a software testing technique in which individual units or components of a software system are tested in isolation from the rest of the system. The goal of unit testing is to verify that each unit or component of the system works as expected, and to identify and isolate any defects or bugs that may exist in the code.

The goal of API testing is to verify that an API (Application Programming Interface) behaves as expected in terms of functionality, reliability, performance, security, and scalability. API testing involves testing the individual components and interactions of an API and ensuring that they conform to the requirements and specifications.

API testing helps to identify bugs, defects, and issues early in the development cycle, reducing the cost and effort of fixing them later.

**Dependencies require for testing**

We use mocha and chai framework for API testing in node js. Mocha is a testing framework that provides a way to write and run tests. It allows you to define test suites, write test cases, and run tests automatically. Mocha is very flexible and can be used with different test runners and assertion libraries.

Chai is an assertion library that provides a way to make assertions about the expected behavior of your code. Chai supports different assertion styles, like "should", "expect", and "assert". Together, Mocha and Chai provide a powerful testing environment for Node.js applications.

**Setup**

**Require step for set up Mocha and Chai:**

1. Install the required dependencies:

Install Mocha and Chai using the below command.

Command : **npm install mocha chai --save**

1. Create a directory and name it “test”
2. Create a test file and include required dependencies.



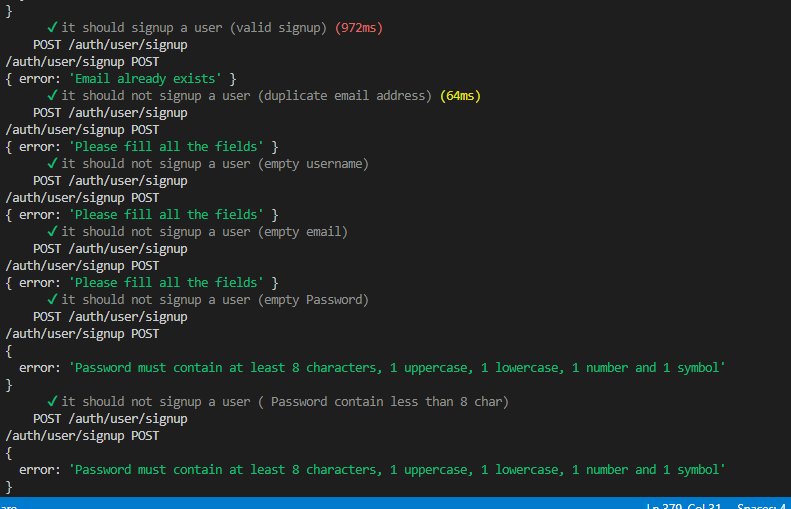
1. Run test cases using below command

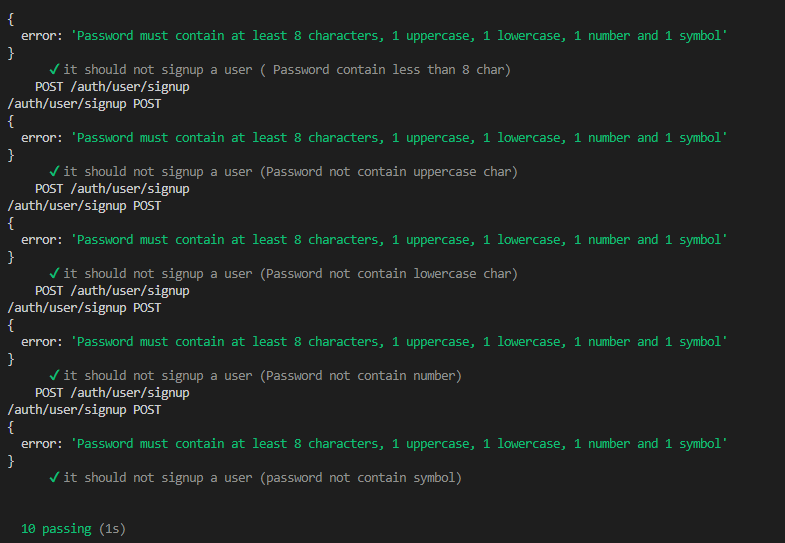
Command: **npm run test**

**Test cases and result**

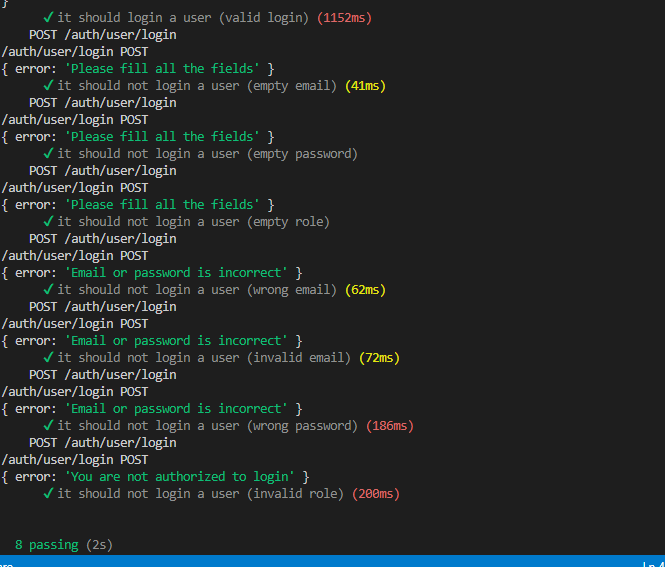
**User api testing:**

**User signup test result** :



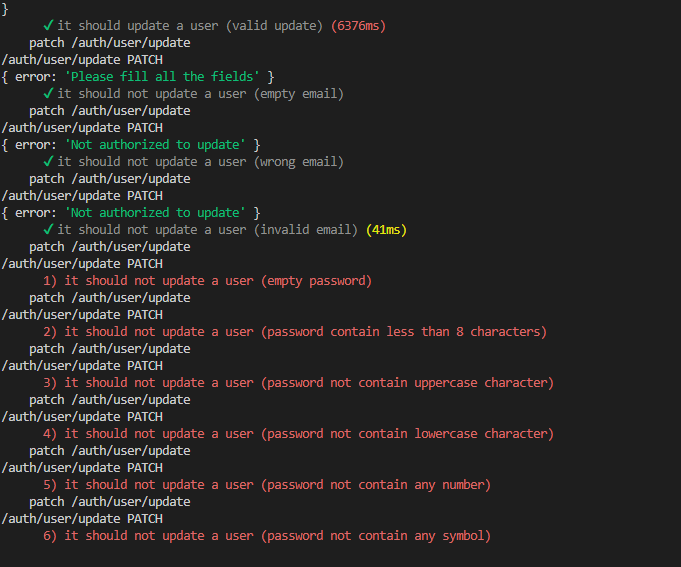


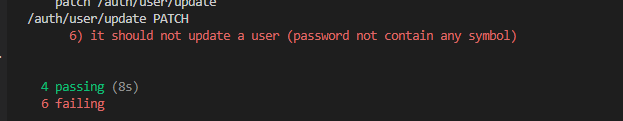
**User login test result** :



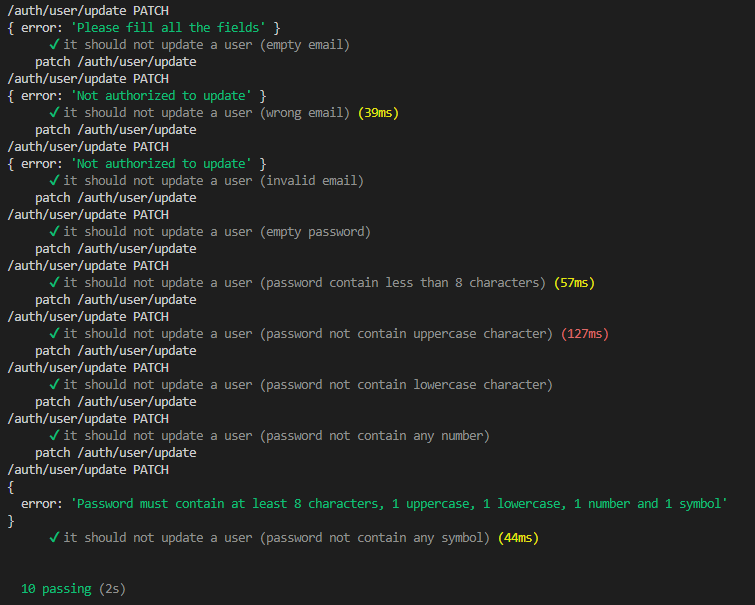
**User update test result (error)**

Passwords contain at least 8 characters , 1 uppercase , 1 lowercase , 1 symbol , 1 number .

****

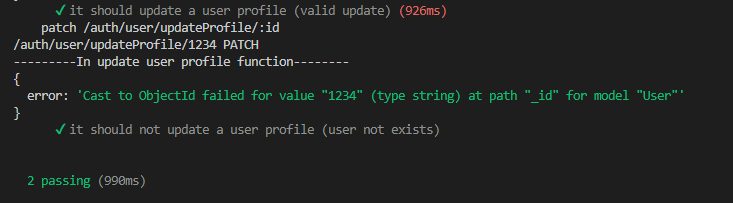
****

After solve the problem

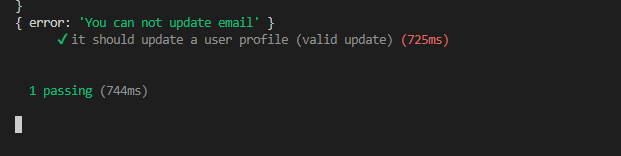
****

**User update profile test result (error)**

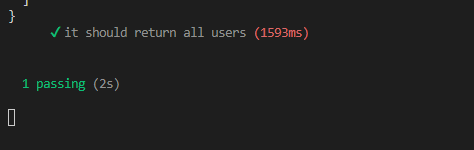
In userprofile update user is allowed to update password ,email also tharegistered\_conferences that should not be happen

****

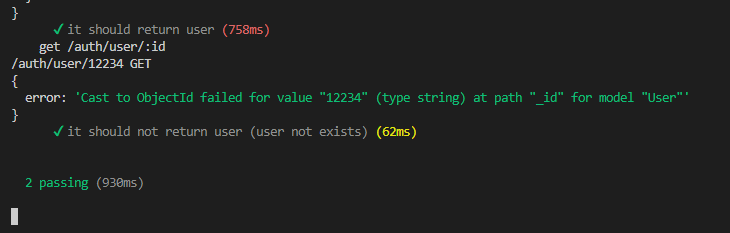
After solve problem

****

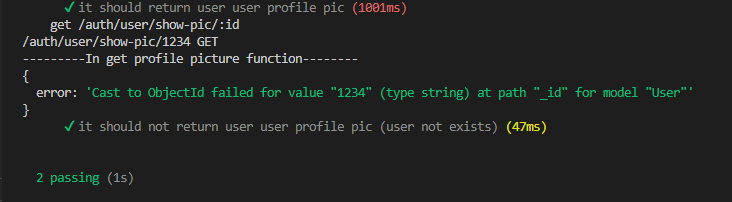
**User getAllUser test result**

****

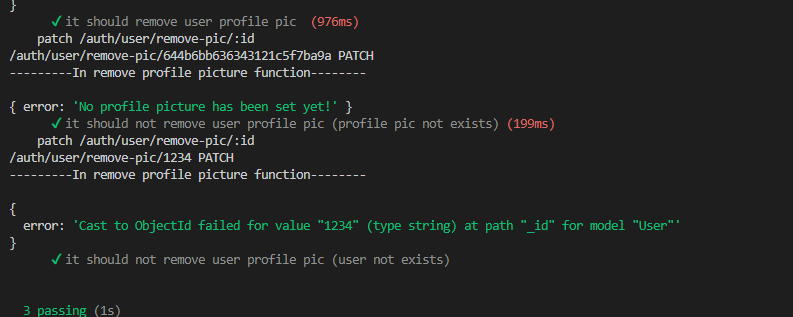
**User getUserById test result**

****

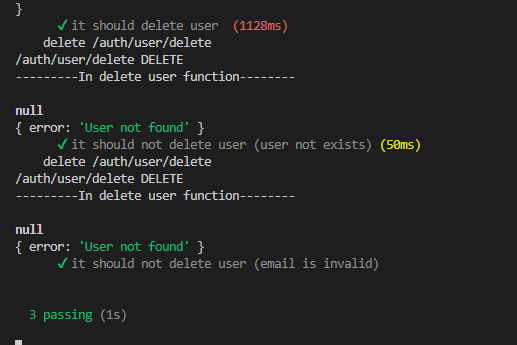
**User get profile pic test result**

****

**User remove profile pic test result**

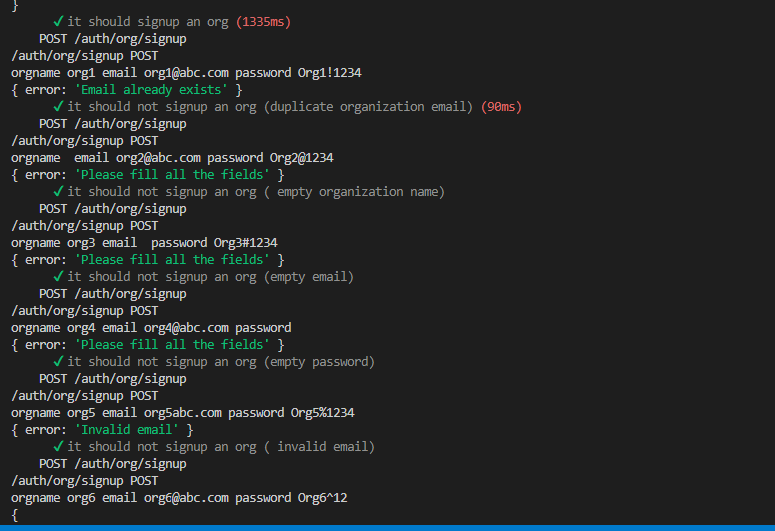
****

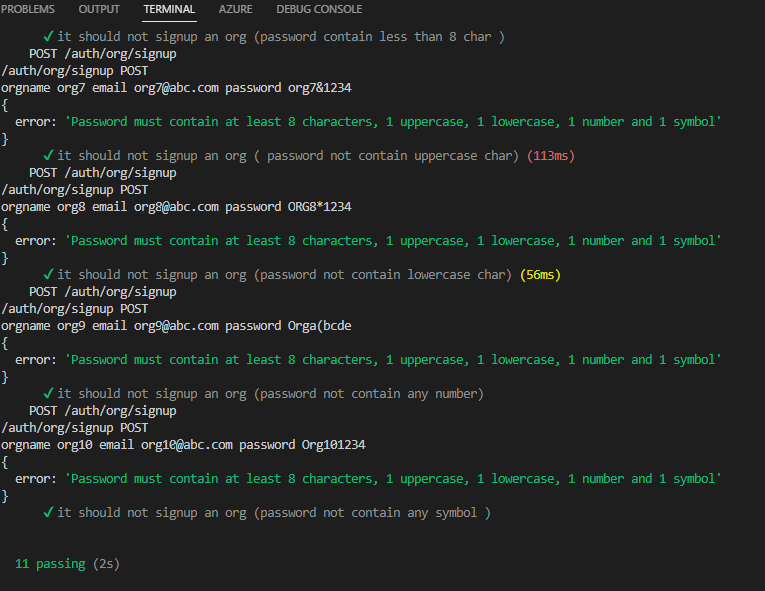
**User delete user test result**

****

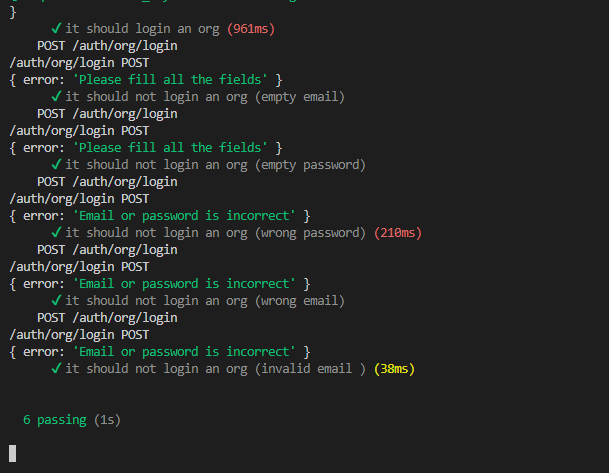
**Organization’s api testing :**

**Organization signup test result**

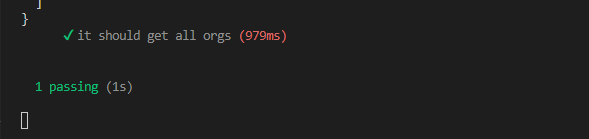
****

****

**Organization login test result**

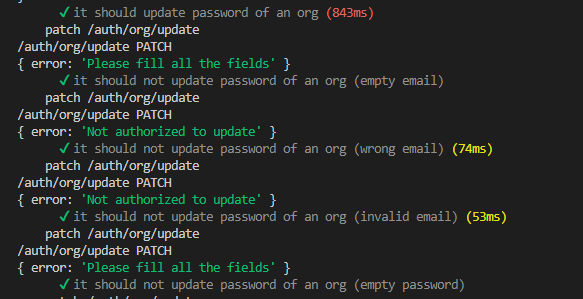
****

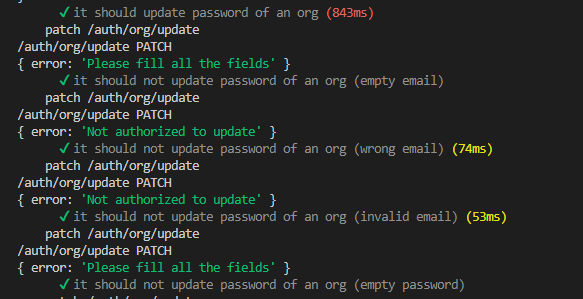
**Organization getAllOrganization test result**

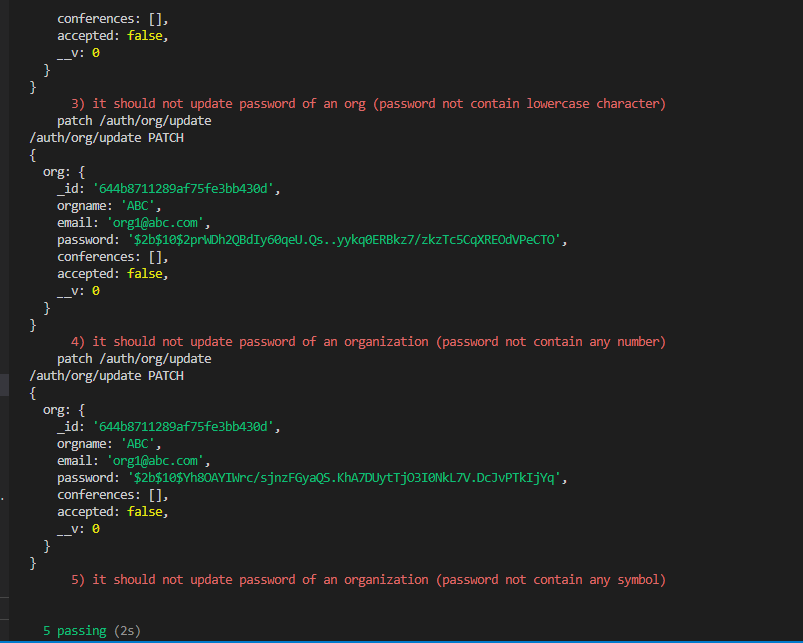
****

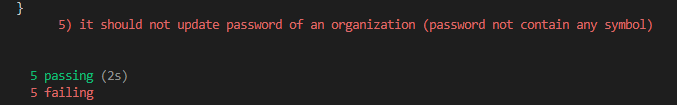
**Organization update test result**

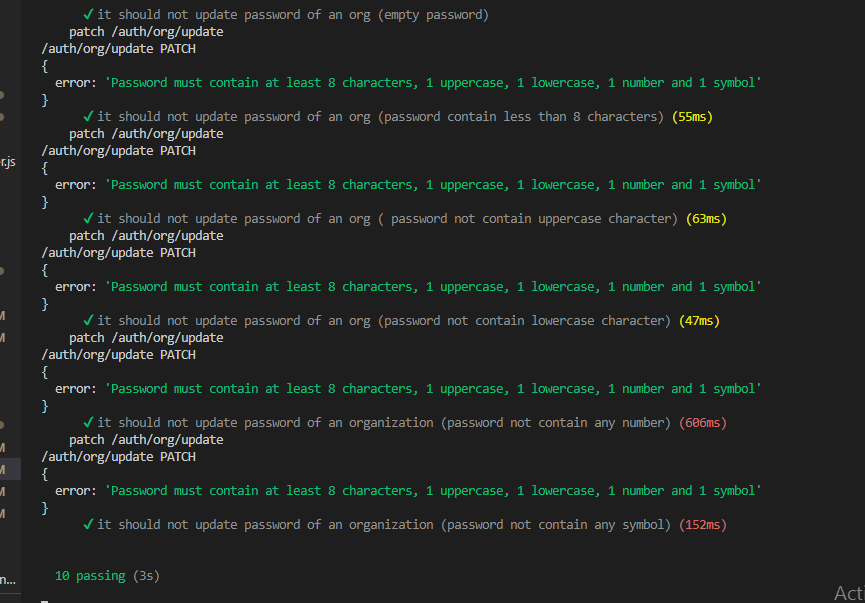
Error : Passwords contain at least 8 characters , 1 uppercase , 1 lowercase , 1 symbol , 1 number .

****

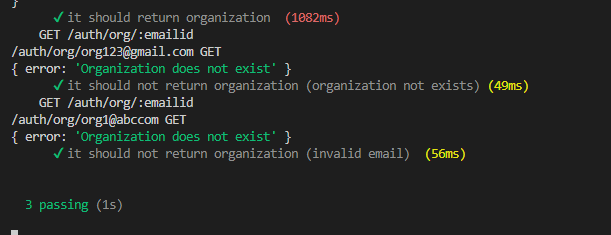
****

****

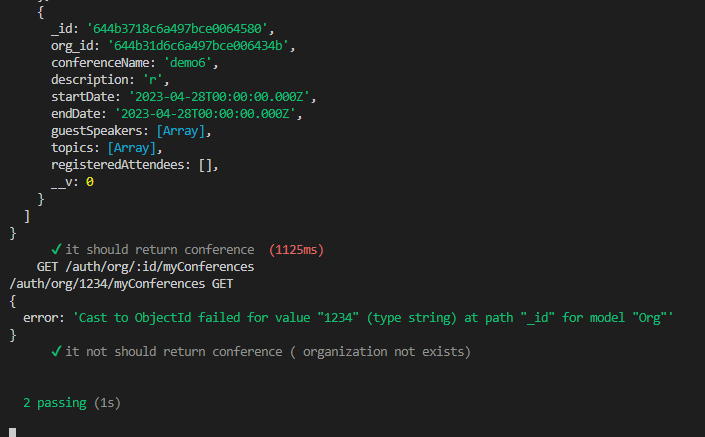
****

**After solve the error **

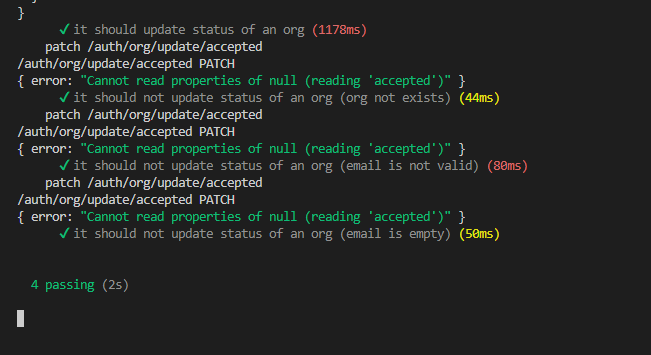
**Organization getOrganizationById test result**

****

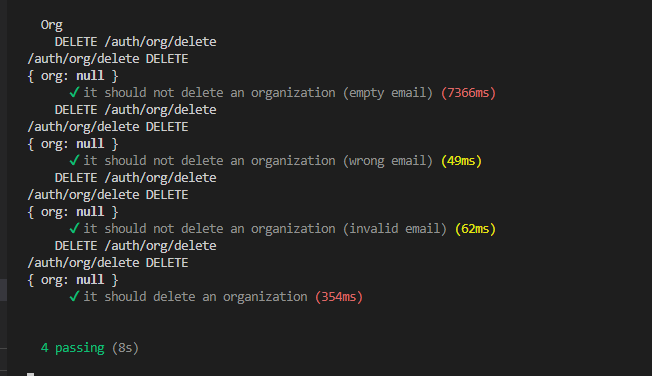
**Organization: get organization's conference by id test result**

****

**Update organization accepted status test result**

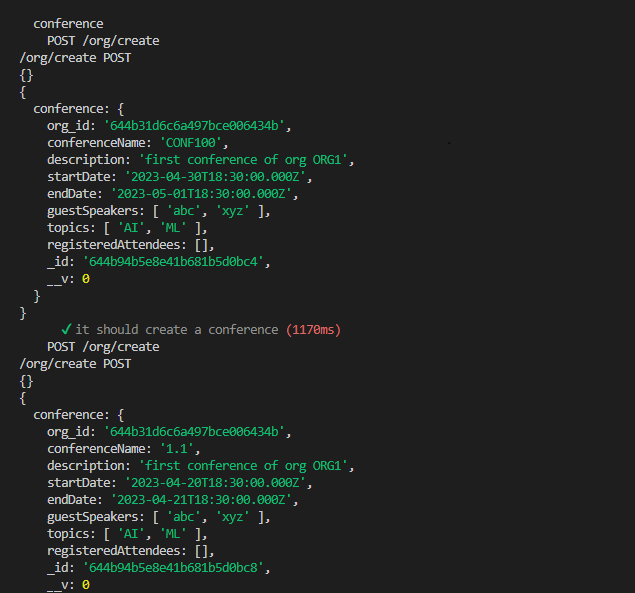
****

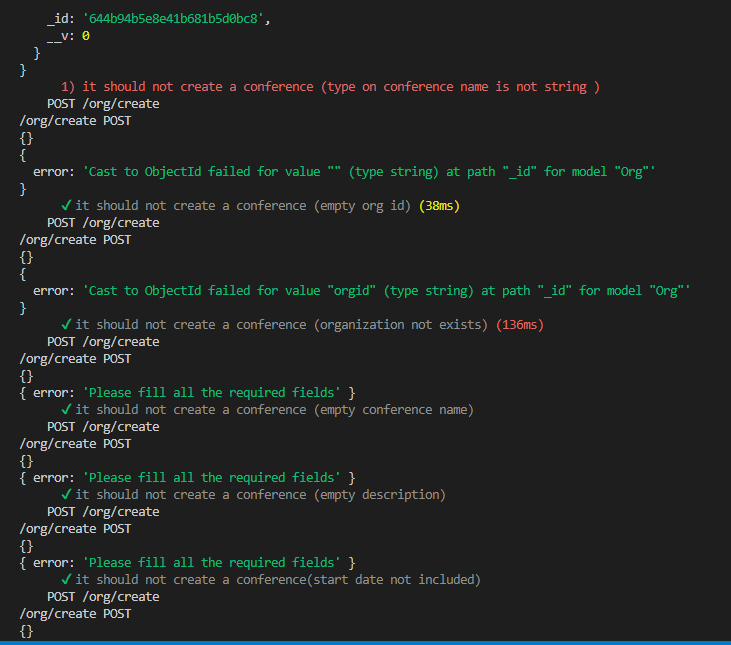
**Delete Organization test result**

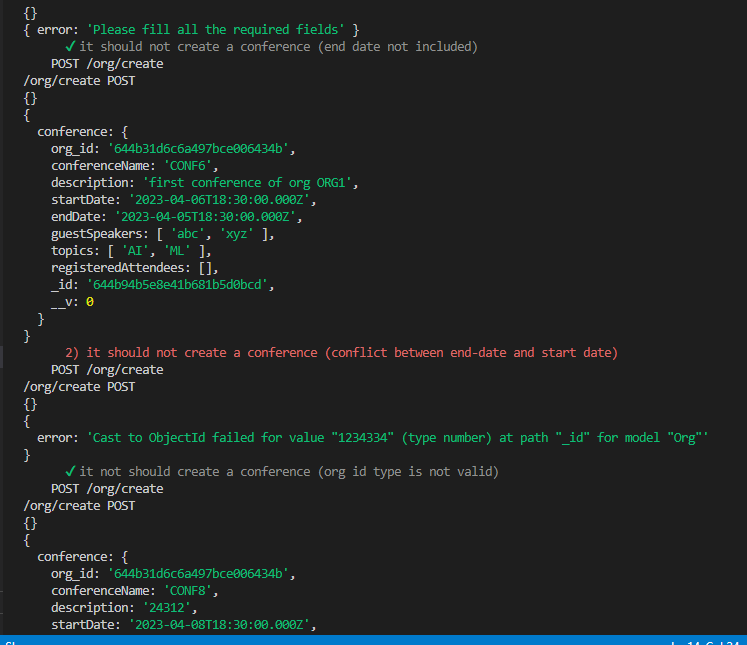
****

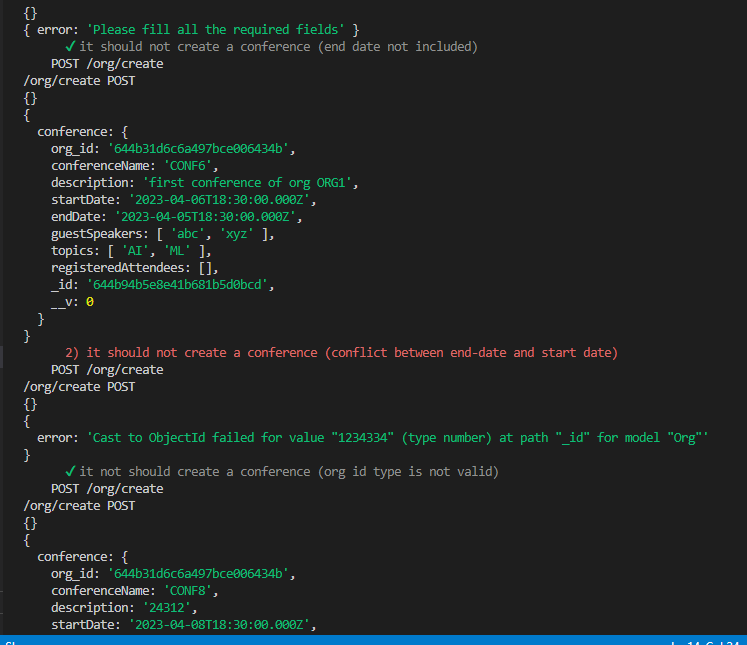
**Conference test result**

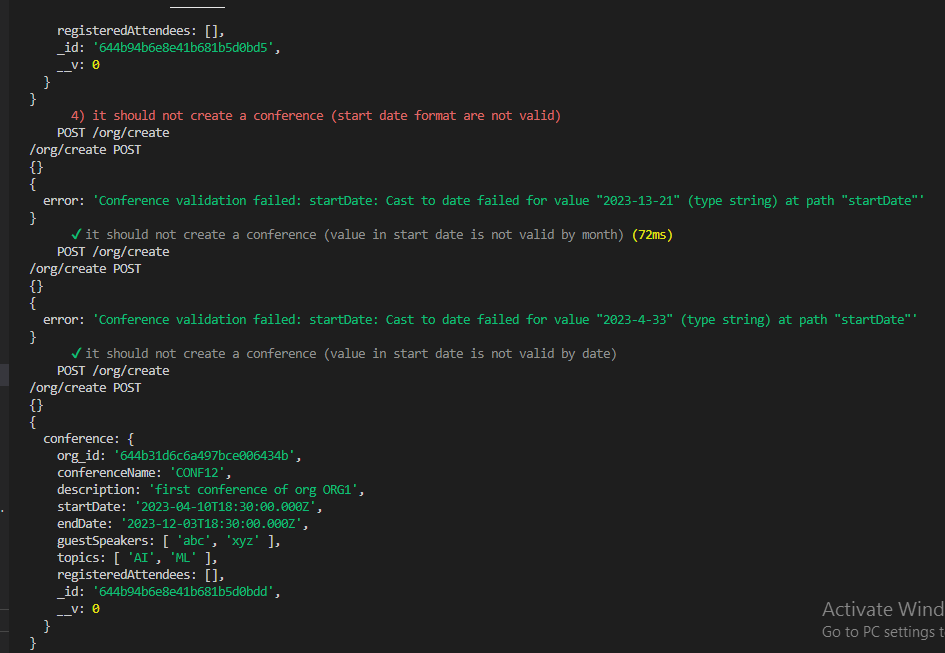
**Signup test result**

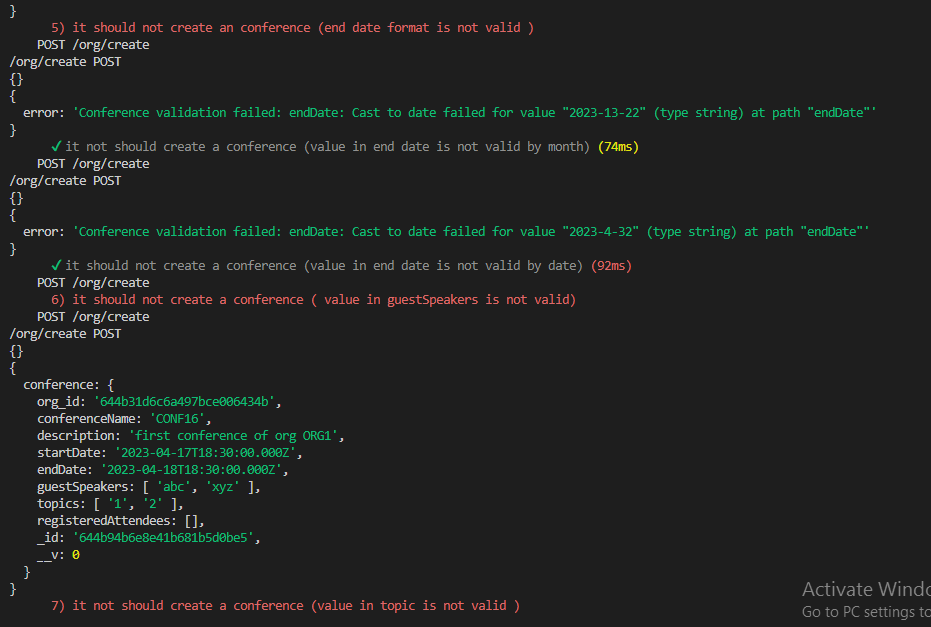


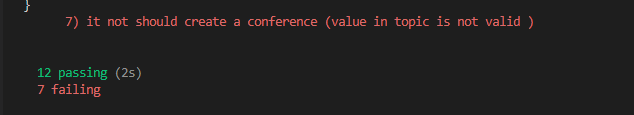




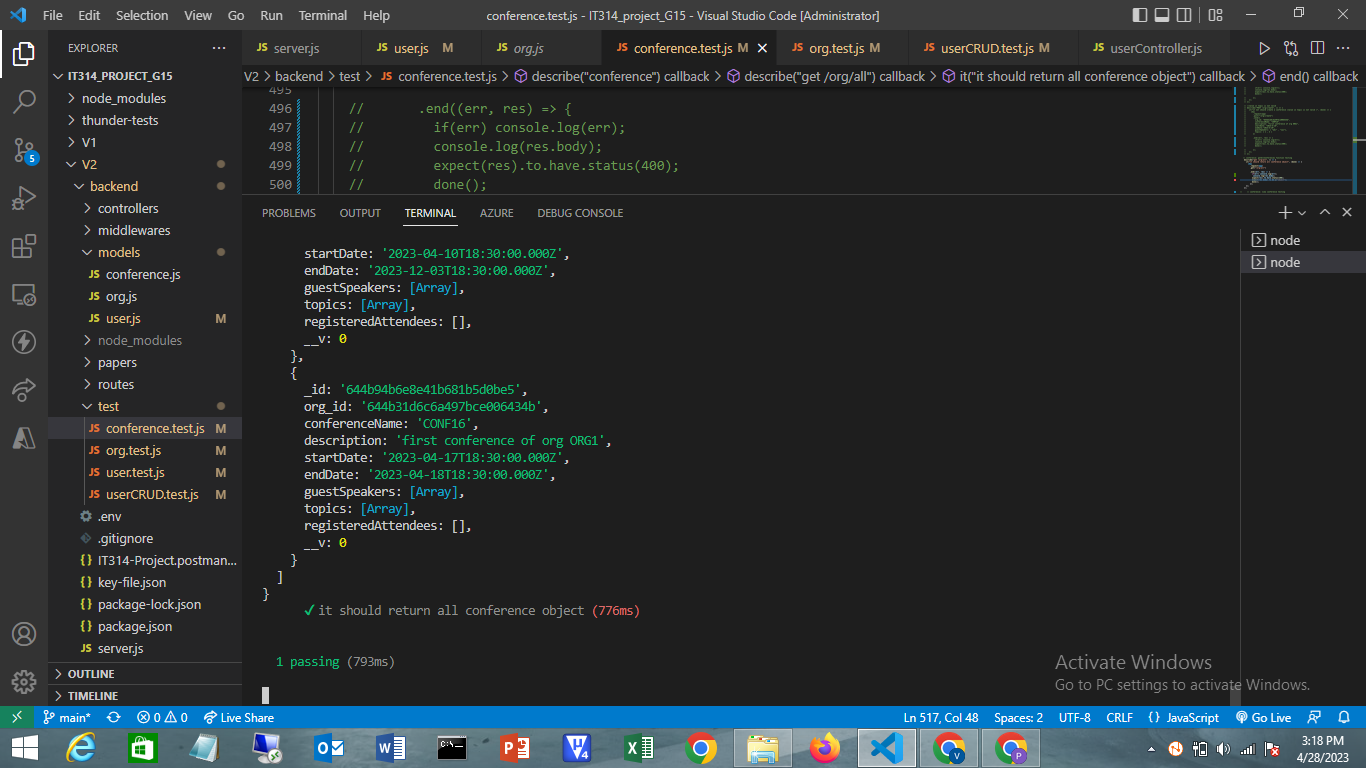






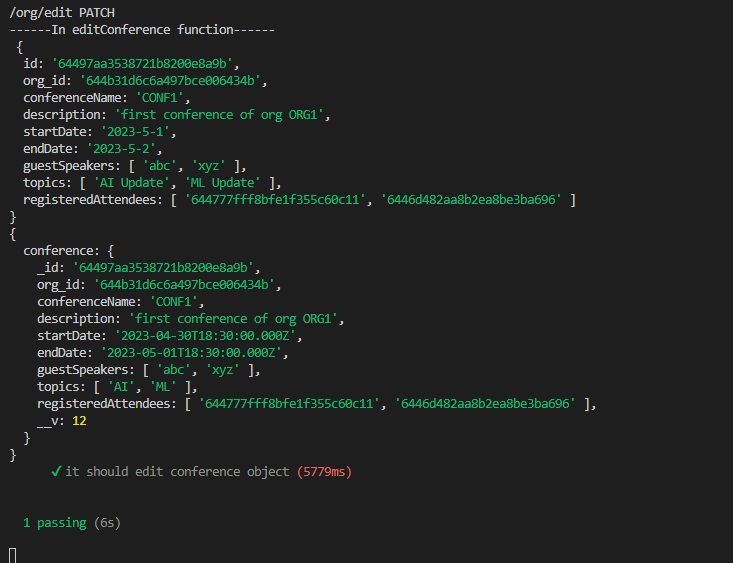


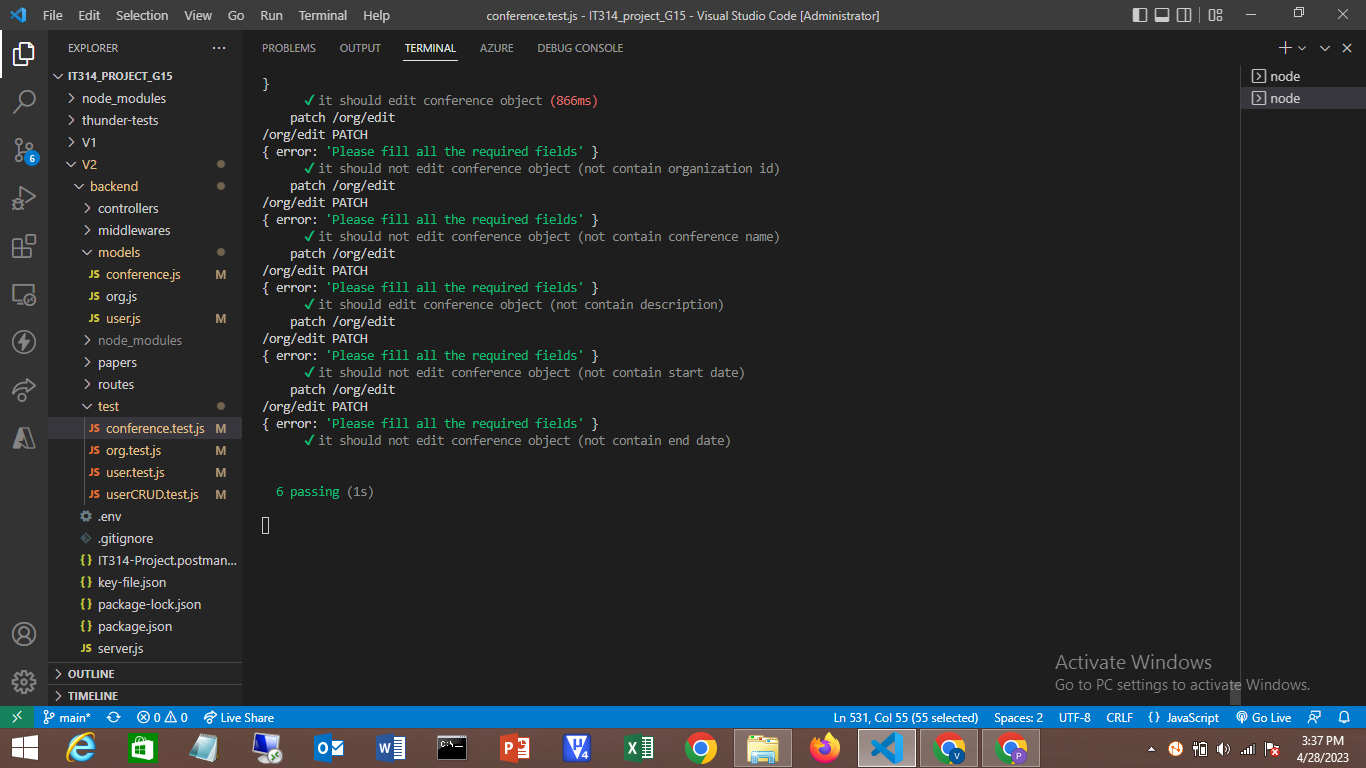
**getAllConferences test result**



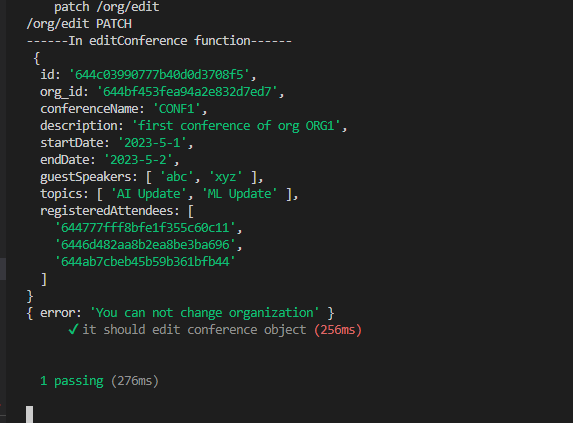
**conference edit function test result**

error : User is allowed to edit registeredAttendees and org id

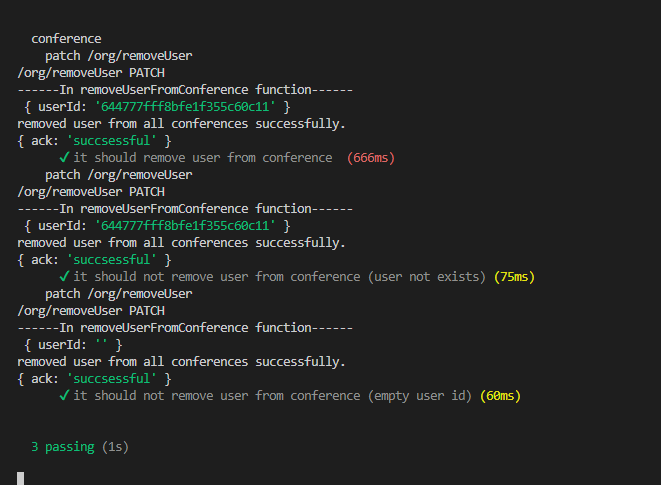




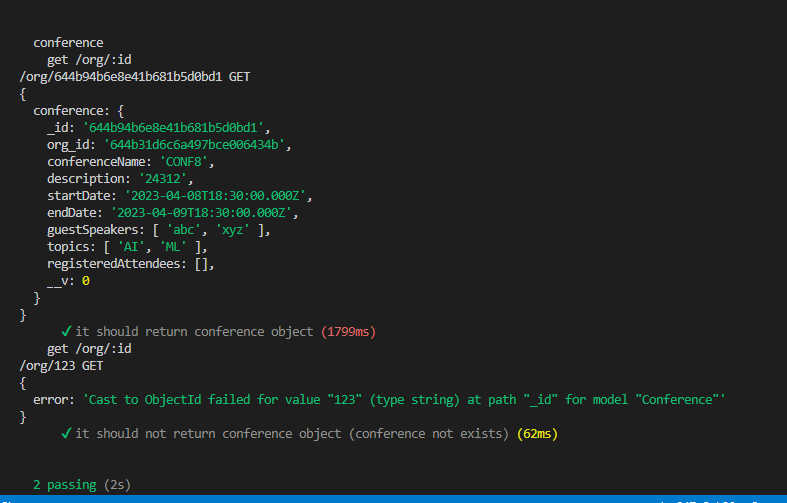
**After solving error**



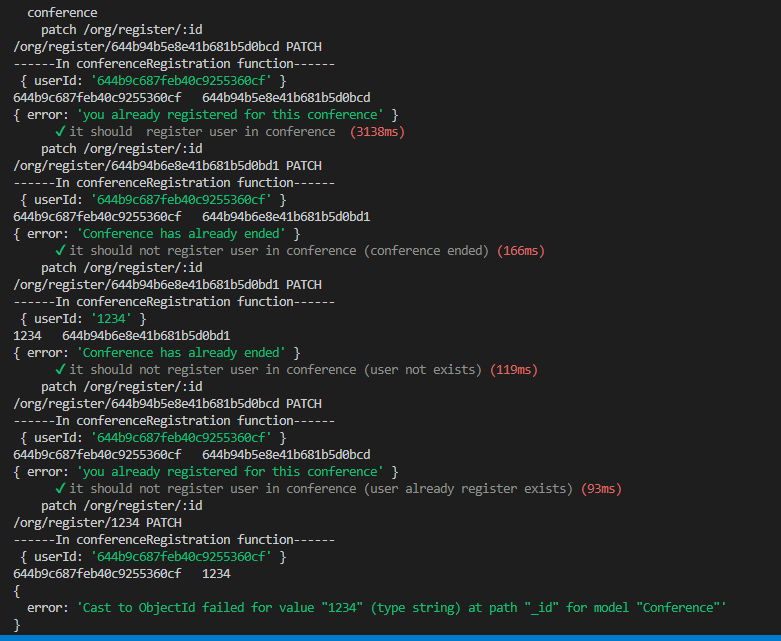
**remove users from conference**

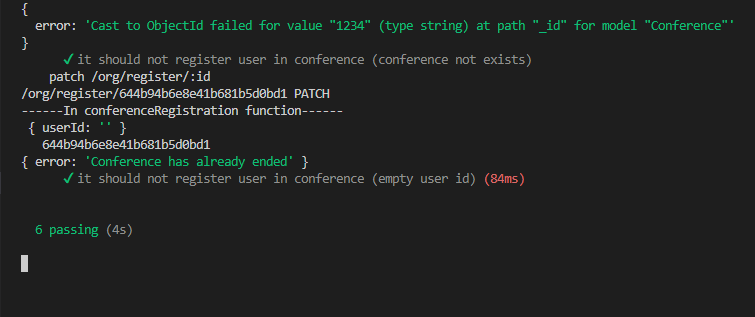
****

**Get conference by Id**

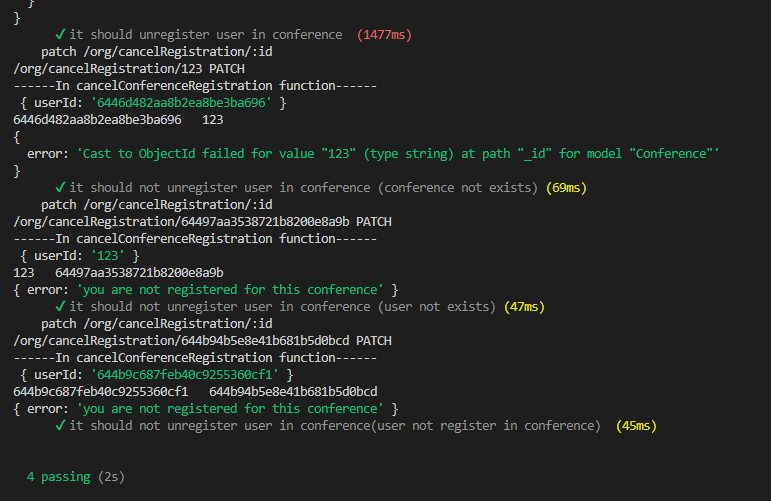
****

**Register user in conference test result**

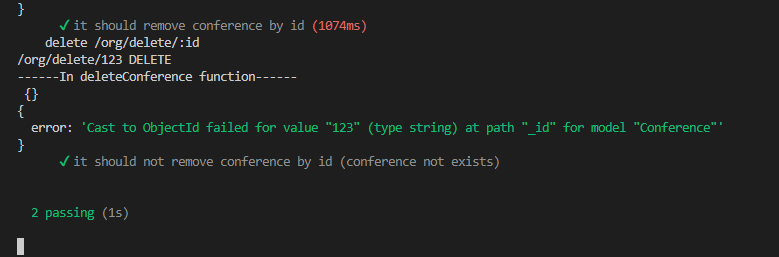
****

****

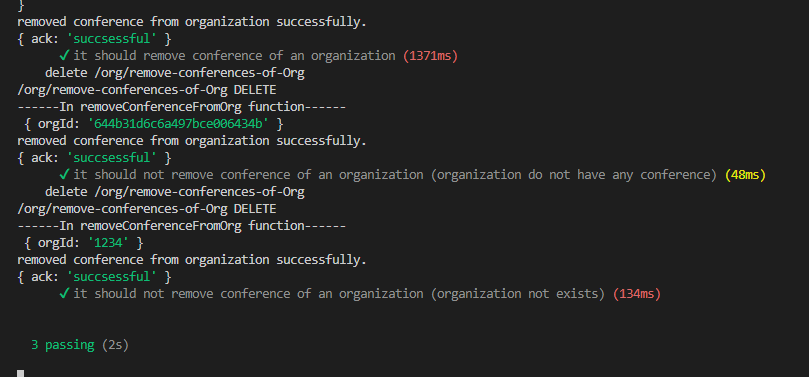
**Conference cancel Registration test result**

****

**conference delete by id test result**

****

**Delete all conference of an organization**

****

**User api test case code** [**link**](https://github.com/MrSMS0000/IT314_project_G15/blob/main/V2/backend/test/user.test.js)

**Organization api test case code** [**link**](https://github.com/MrSMS0000/IT314_project_G15/blob/main/V2/backend/test/org.test.js)

**Conference api test case code** [**link**](https://github.com/MrSMS0000/IT314_project_G15/blob/main/V2/backend/test/org.test.js)