Integration Test #1 - Simulating Database Error for Registration Endpoint:

- This integration test checks if the registration endpoint handles database issues properly. It starts by setting up valid user registration data to simulate someone signing up. It then intentionally triggers a database error using a mock function, which makes the database connection fail when the registration process runs. The goal of this test is to make sure the app doesn't crash when the database fails, the user gets a clear error message with the correct status code, and the error is logged for debugging but no sensitive info is shown to the user.

Integration Test #2 - Registration and User Login Pipeline

- This integration test checks if the registration and login components work together as intended. When a user registers a new account, they should immediately be able to successfully login using the same credentials as their registered account. This test simulates the post requests made for registration and login together to ensure that the components work together to successfully log in a user account into the application after registering that account.

Integration Test #3 - Cancel Booking

- This integration test validates the endpoint functionalities of the cancel booking feature. It checks that users can successfully cancel active bookings and that the system responds appropriately in different scenarios without crashing. When a booking is successfully canceled, the test verifies that the booking status is updated in the database, and removed from the active booking list. A success message is displayed to the user. It also handles cases where an attempt to cancel a non-existent booking is made or when the list is empty, and an error message is displayed. Also confirms that the user interface reflects backend changes accurately and that after cancellation, the booking is not shown on the cancel booking page.