Buy Feature Test Plan

- For test cases, there would be 2 users, each with an account, connected payment information (paypal or something), and one user (User 2) has a ticket on their account (with a ticket id).
- User 1 goes to the buy page, and sees User 2's ticket available for purchase. User 1 clicks the buy button on User 2's ticket, and is redirected to a payment portal. After the third party payment is accepted, User 2's ticket moves to user 1's account. User 2 is then asked to transfer the ticket before they receive the money for user 1's ticket. If the deadline is not met, then User 2 would receive a potential fine or some sort of penalty.
- This would involve accounts interacting with each other, which would be a cloud test environment.
- The results of this test:
  - Positive test case example, transaction goes through. We would be able to query the database and see the ticket on User 1's account. User 2 would receive the email to transfer, User 2 is able to transfer the ticket.
  - Negative test case example, transaction fails. User 1 clicks buy, User 2 doesn't transfer in time, and receives the penalty. Query shows User 2 still has the ticket, as the transfer didn't go through.
- User acceptance testers would be able to navigate the buy page, buy, and through a variety of different interactions, would be able to complete or fail transactions, and then take notes to show the different possibilities of the buy page working or not working.

Sell Feature Test Plan

- For the sell feature, there would be just 1 user, trying to sell their ticket. They would sell their ticket associated with their account for a certain event for a price of their choosing onto the website.
- The environment for this test would be localhost, as there is only one user involved.
- The results of the test would be as follows:
  - Positive Test Case:
    - User sells a ticket, and it appears on the listings for the event for any other user or themselves. They would see that their ticket is being sold and if queried, the ticket would still be under the user's account, but it would also have a for sale flag.
  - Negative Test Case:
    - User tries to sell a ticket, but it is either not under their account, for an old event and not eligible for selling, or some other various error has occurred. The user would not be able to sell a ticket if they do not own it– the option wouldn't be available to them. Upon clicking sell, the user would be prompted with a message saying they have no tickets, and they would have to upload information about their ticket to the website

- User acceptance testers would see tickets on their account, and would click sell for a certain event, and would see the listing of their ticket for the price that they chose on the website. If a tester did not own a ticket, they would see the message if they tried to sell.

Transfer Feature Test Plan

The ticket transfer feature will be tested on localhost:3000 using Chrome and Firefox browsers, connecting to our PostgreSQL test database and CU Boulder's Shibboleth SSO test environment. Testing will be conducted by three CU Boulder students (one experienced with ticket transfers, one new to the process, and one regular user) plus one test administrator from our development team. We will use test data including a CU vs Nebraska Football ticket dated September 15, 2024, and multiple @colorado.edu test accounts with pre-populated purchase history. Three main test cases will be executed: 1) A standard ticket transfer where Student A logs in, navigates to My Tickets, selects a ticket, and transfers it to Student B's email, verifying ownership updates and notifications; 2) Transfer acceptance where Student B logs in, views pending transfers, and accepts the ticket, confirming proper ownership change and account updates; and 3) Invalid transfer attempts including transfers to non-colorado.edu emails and already-transferred tickets to verify proper error handling. Success criteria requires all test cases to pass, transfers to complete within 30 seconds, email notifications to function correctly, database integrity to be maintained, and no security vulnerabilities to be found. Testing will occur in the Engineering Center computer lab over two days, with final sign-off requiring successful completion of all test cases and thorough documentation of results.