

Who: Filip, Sneha, Mackenzie, Kelly, Sultan, Alex

Project Name: Geotaste

User Acceptance Tests:

Three Features to be Tested:

- Login Page
 - Unit testing
 - Testing whether the name field is right
 - Testing whether the password that they imputed and the confirm password is right

```
describe('Test form validation', () => {  
  
  it ('number in Name field is invalid', ())  
  {  
    let valid = validateName('Sneha123');  
    expect(valid).to.be.false;  
  }  
});  
  
describe('Test form validation', () => {  
  
  it ('other symbols are invalid in the name field', ())  
  {  
    let valid = validateName('Sneha!@#');  
    expect(valid).to.be.false;  
  }  
});
```

- Country page integration
 - Unit Testing
 - Testing the map sends the proper form data by making sure the content and format of the url's function as expected for all implemented countries
 - Testing that the page properly renders the information provided by the JSON by loading the page and making sure the information is accurate and displayed correctly
 - Testing that the country page properly loads the dynamically generated elements for the song list by using the data from the database to generate the song list and checking for accuracy

- Integration testing
 - Testing that the map properly sends data retrieved in the get request that renders the country page by opening the map settings and checking the urls send the correct information in the proper format
 - Testing that the country page properly loads the data from the JSON file to render the title, image, etc. sending an api request to render the page with a given country id
 - Testing that the database query pulls the desired result from the database by running a query on my local database
 - Testing that the song list sends the proper form data to load the song page by reviewing form data and cross-checking it with the way we render the song page

```

1  const chai = require("chai");
2  // Chai HTTP provides an interface for live integration testing of the API's.
3  const chaiHttp = require("chai-http");
4  chai.should();
5  chai.use(chaiHttp);
6  const { assert, expect } = chai;
7
8  const itParam = require('mocha-param').itParam;
9
10 describe('Map data validation', () => {
11   it ('map sends correct country id', (done) => {
12     chai
13       .request(server)
14       .get("/country/:id")
15       .send({country: 'geotaste.herokuapp.com/country/belize'})
16       .end((err, res) => {
17         expect(res).to.have.property("country_id").equals("country");
18         done();
19       });
20   });
21 });

```

- Music Page integration
 - Unit Testing
 - Test that music data is properly pulled from a database and displayed
 - Test Spotify id is pulled and displays proper preview
 - Integration Testing
 - Test song is from correct country
 - User Acceptance Testing
 - Proper discogs id is pulled
 - Once discogs id is received, all relevant information is displayed

```
//positive case, correct id is sent
it("should be valid discogs_id", (done) => {
  chai
    .request(server)
    .get("/music")
    .end((err, res) => {
      expect(res).to.have.status(200);
      expect(res.body).to.be.an('int');
      expect(res.body).to.have.property('song_choice');
      //or song_id or discogs_id, whatever we choose
      expect(res.body.discogs_id).to.be.an(Number);
      expect(res.body.discogs_id).to.be.greaterThan(100000);
      done();
    });
});
```

Individual Contributions:

- Sneha
 - Worked on the login page and created a database to store the username and the password, email, first name and last name
 - Latest git commit - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-012-03/commit/4480e0cc1ec51fe4cc582d76d78d7c3818c13f41>
- Mackenzie
 - Worked on the country page, updated the code for the song list to create an actionable scrollable list. Began testing the html for the page, because I'm using some unfamiliar classes.
 - Latest git commit: <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-012-03/commit/d313264083975c1741bbed6ed4caf156e71033e2>
- Filip
 - Finished the database and hosted it on heroku
 - Latest git commit:
 - <https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-012-03/commit/fe58f95585d3a5ecac77c85a0e1ddb8127599199>
- Kelly
 - Worked on completing the country.json file and started testing to see if the database filip created is able to work and be used by the country page. Also started working with filip and mackenzie on getting the data from the countries.json file to display on our website.

- Latest git commit:
<https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-012-03/commit/6d2670a1c99b71c42567b80c27c2f9bc4b8d4131>
- Sultan
 - Worked on Music history and developed all the required javascript files. Also, created a developer app in Spotify to use the client id, client secret, and added the call back in Redirect URIs.
 - Latest git commit
<https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-012-03/commit/2c06f3f26746bbf65d3aaf53602ca487a37d7581>
- Alex
 - Worked on the music page and integrated it with our database. Uses discogs id to retrieve relevant data for the page.
 - Latest git commit:
<https://github.com/CU-CSCI-3308-Fall-2021/CSCI-3308-Fall21-012-03/commit/2e30b484572386cb2b1967aee86c3d52fd86b803>

Jira Screenshot:

The screenshot shows a Jira Planning Sprint board for the project 'DreamTeam-012-03'. The sprint is titled 'Planning Sprint' with the description 'Research how integrations will work and plan out application architecture'. It has 0 days remaining. The board is organized into columns: 'TO DO', 'IN PROGRESS 1 ISSUE', and 'DONE 6 ISSUES'. There are two epic items: 'D00-3 Website UI' (3 issues) and 'D00-4 User Profile' (1 issue). The 'D00-3 Website UI' epic is expanded, showing three issues in the 'TO DO' column: 'User wants to link application to main streaming services' (D00-17, 6 votes, assigned to FC), 'User wants to see interpreted data from API' (D00-29, 5 votes, assigned to AP), and 'User wants to link with Discogs API (Architecture)' (D00-28, 4 votes, assigned to ME). The 'D00-4 User Profile' epic is also expanded, showing one issue: 'User wants to be able to login'.

Planning Sprint

Research how integrations will work and plan out application architecture

   0 days remaining

SY

FC

KW

AP

+2



Epic ▾

GROUP BY ▾

TO DO		IN PROGRESS 1 ISSUE	DONE 6 ISSUES ✓	+
▼ ⚡ D00-4 User Profile 1 issue TO DO				
			User wants to be able to login D00-19 ✓ 1 SY	
▼ ⚡ D00-12 Geo Origin 1 issue TO DO				
			Spotify Music Preview D00-25 ✓ 5 SA	
▼ ⚡ D00-26 UI Map 2 issues TO DO				
		UI integrating the map and the user should be able to listen to the music when they click on different parts of the map D00-24 2 KW	Setting up Location-Label Database D00-23 ✓ 3 FC	