**Connection Management System**

**Github link:**

[**https://github.com/AndyLeUNCC/eventsCaseStudy**](https://github.com/AndyLeUNCC/eventsCaseStudy)

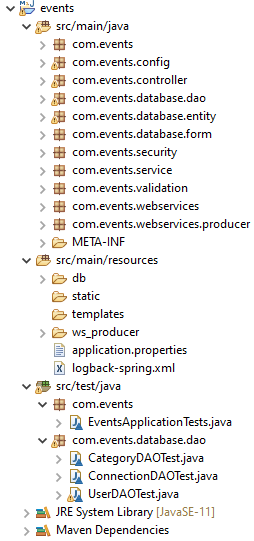
**Project Description**

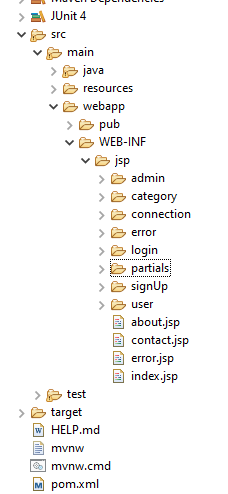
My project is to build a website system that allows users to create events to connect with other users on the internet. The system allows website visitors to view all events created by registered users. But they cannot register to attend if they are not a registered member with an account on the system.

**Summary of Pages**

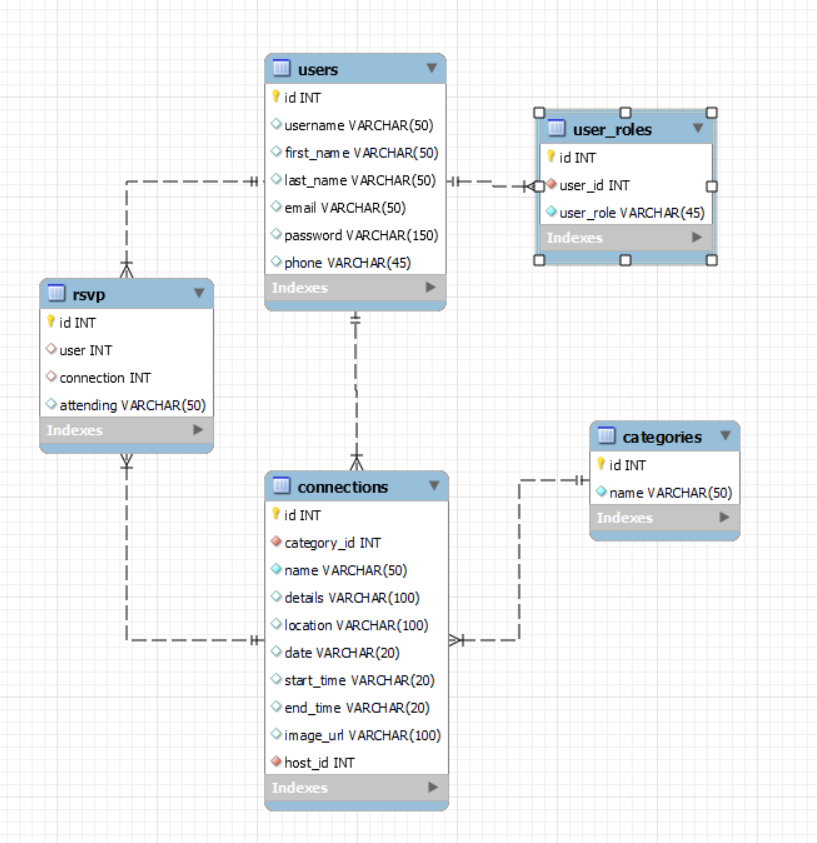
* Home Page contains links for user-specific navigation such as login, signup,... The main content area provides an introduction and overview of the site.
* User Profile: contains information about the events created by that user and the RSVP that the user chose.
* Connections Page: provide all of the connections/events created by other users. Each category contains at least one connection/event.
* Connection Detail: provide the description of a detailed connection, such as category/topic, host user name, time date, location.
* New Connection: allow users to create a new connection.
* Login page: allows users to log in to the site
* Signup Page: This allows users to create a new account for the site’s members to get permission to create, update their own connections.
* Search Page: allow the user to search the connection/event based on the name of the event and the categories list of events
* About Page: provide the visitors with some information about the site.
* Contact Page: provide the contact information
* Admin Page: allow the administrators to maintain the site’s members such as update, delete the members.

**Project Structure**



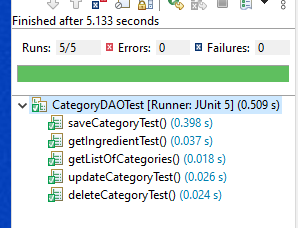


**Database Diagram**

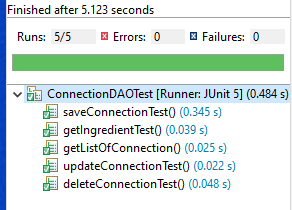


Junit Testcases Evidences

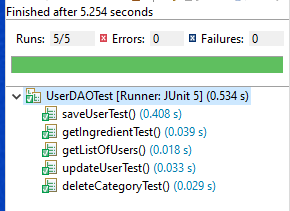
1. CategoryDAOTest



1. ConnectionDAOTest



1. UserDAOTest



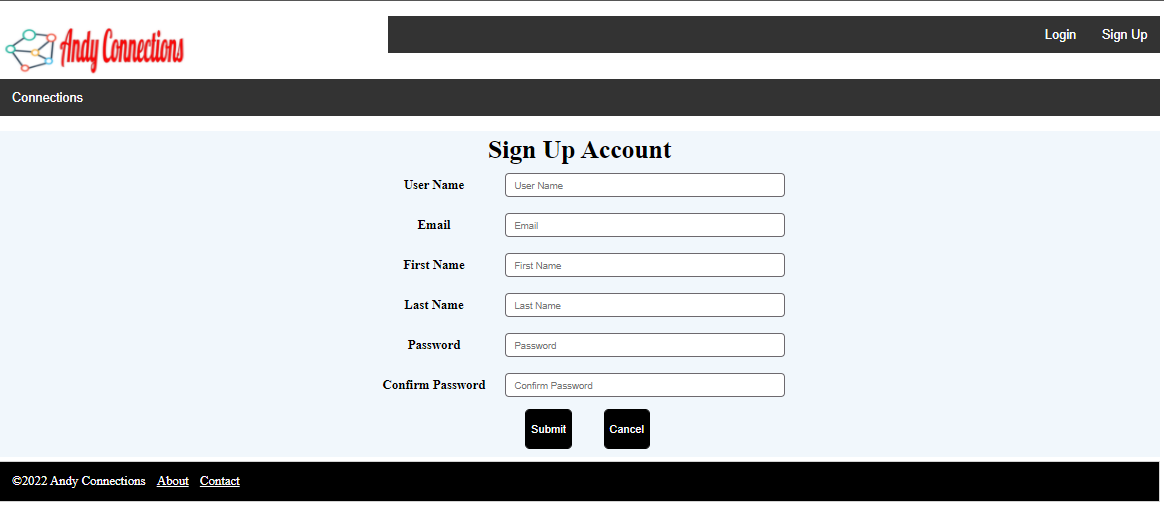
Main Pages in Web Application

1. Home Page

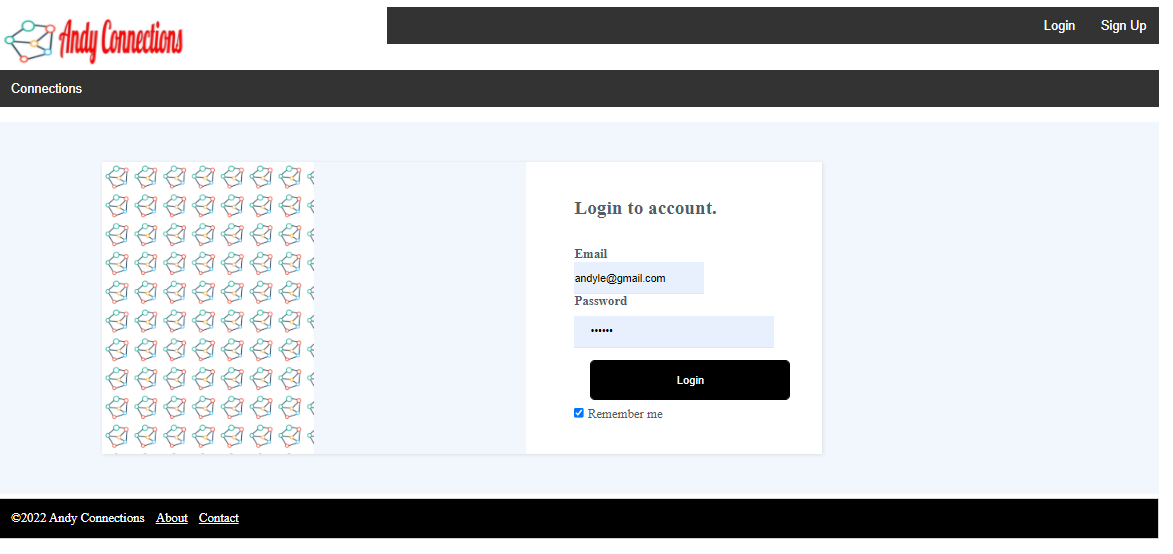
Using slider functionality to make the home page dynamically and vividly.



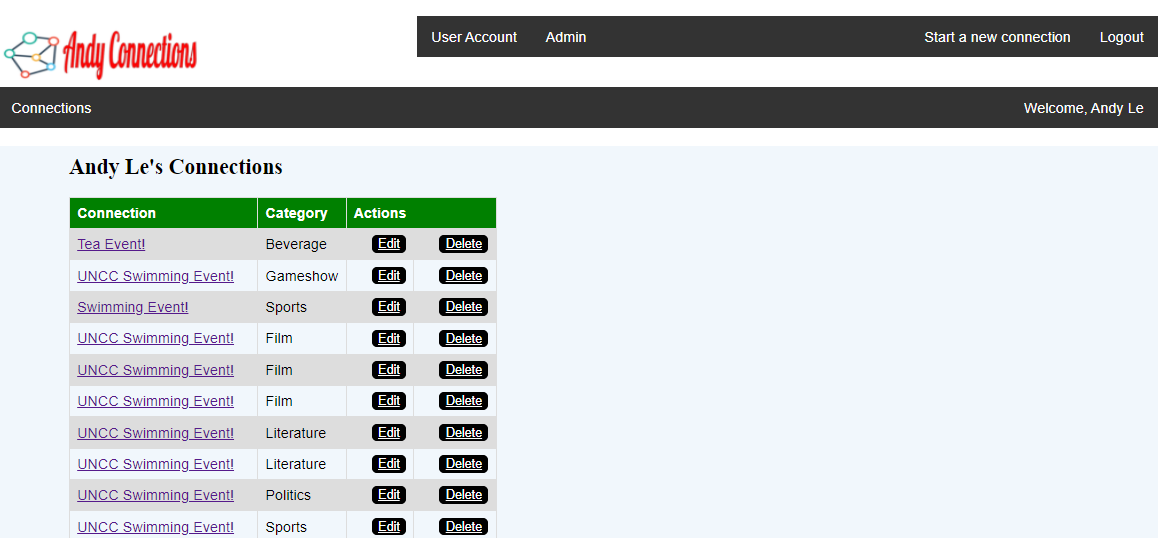
1. Sign Up Page



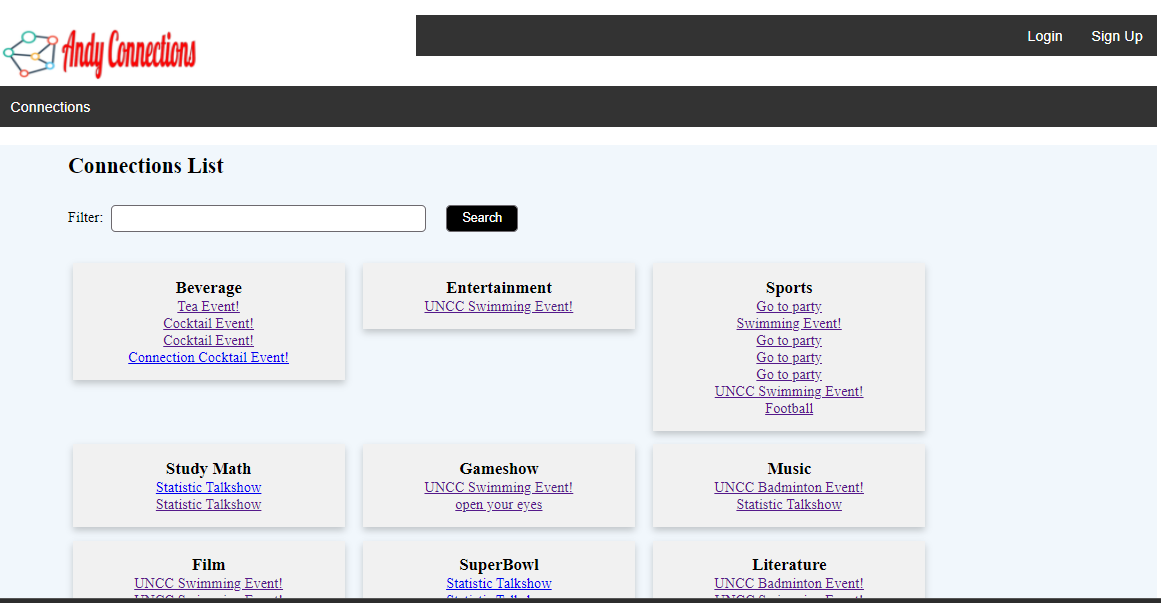
1. Login Page



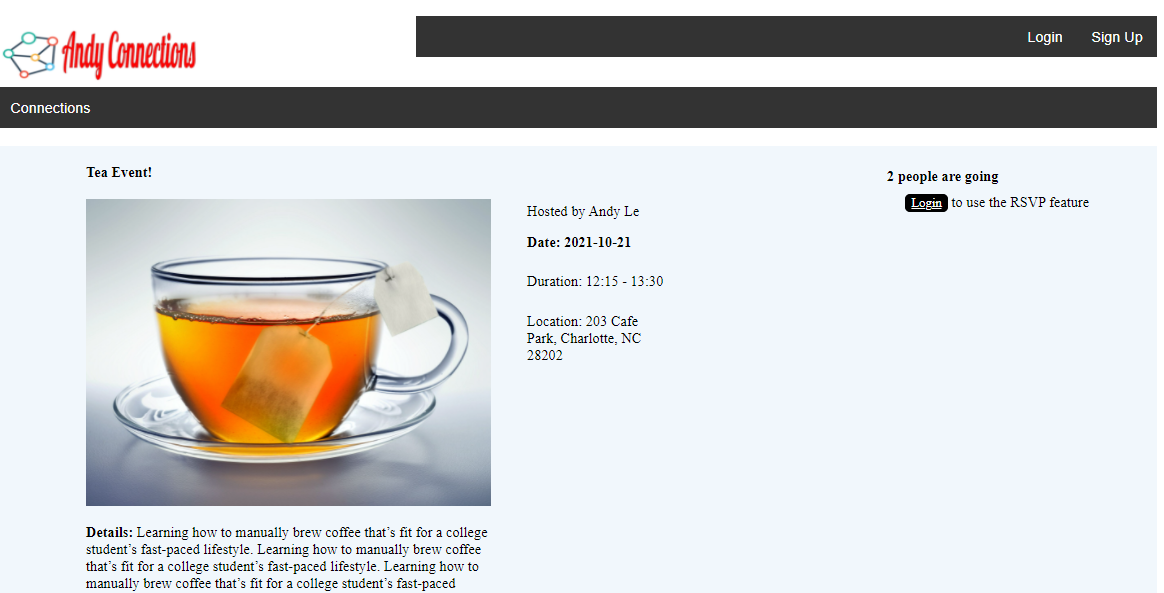
1. User Profile Page



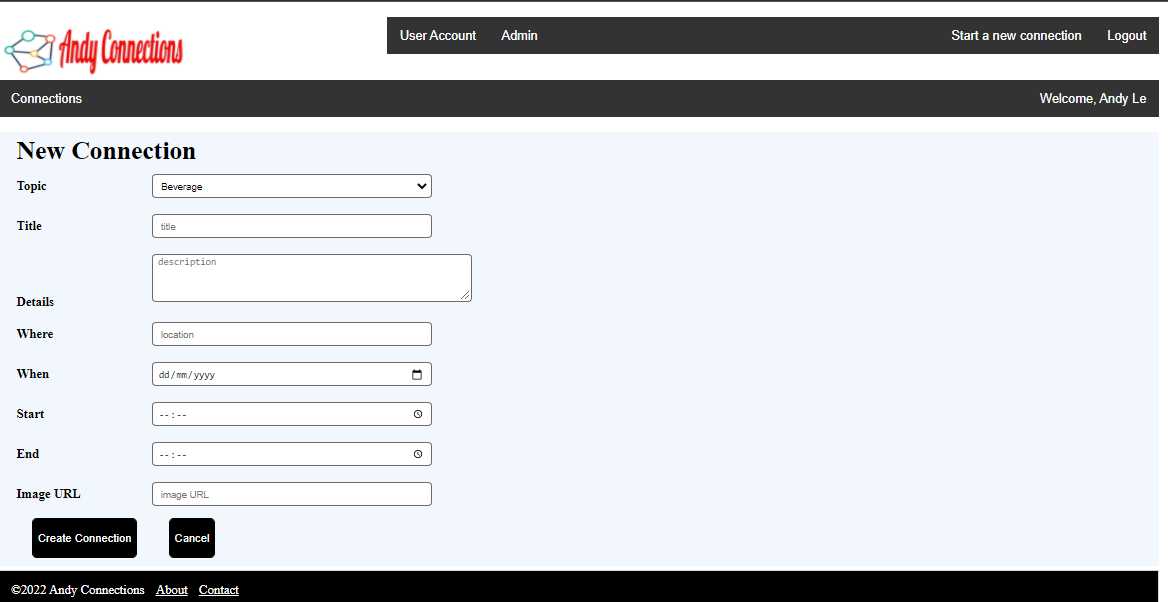
1. Connections Page



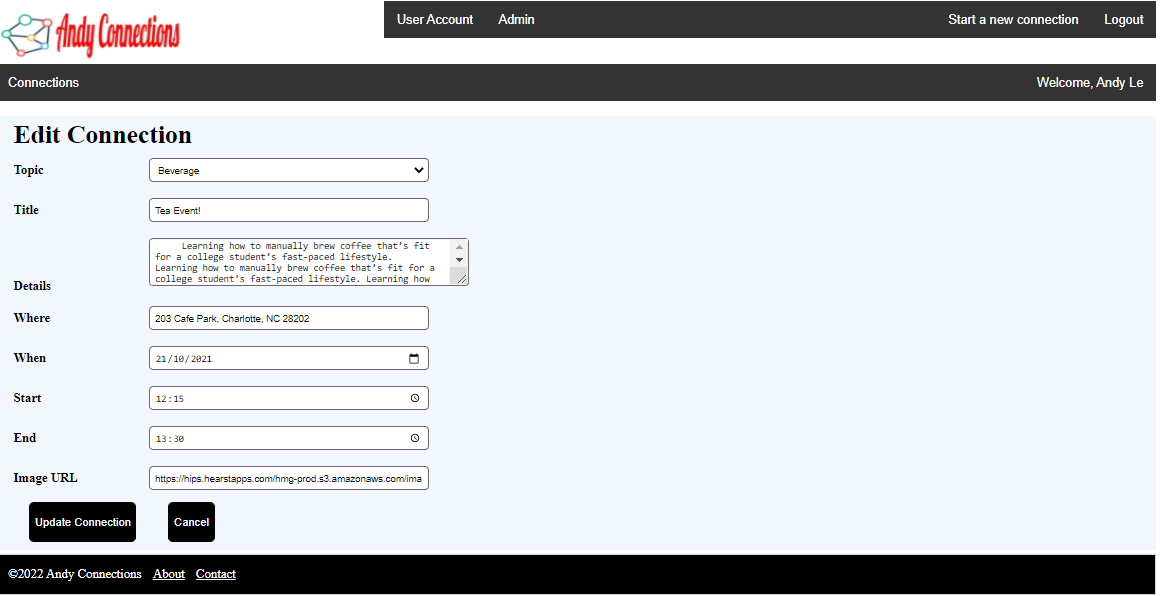
1. Connection Detail Page



1. Create a new Connection Page



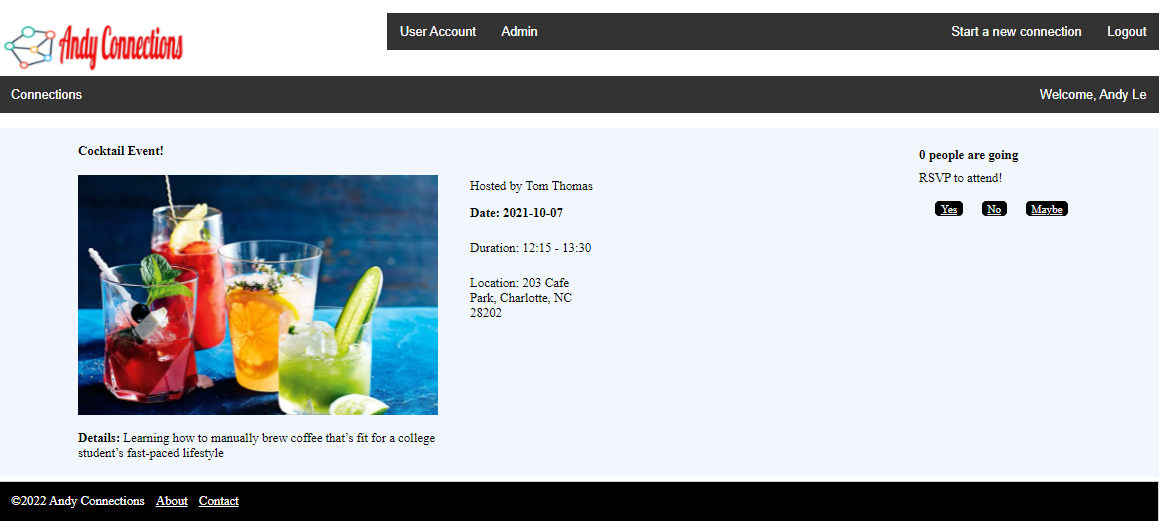
1. Edit Connection



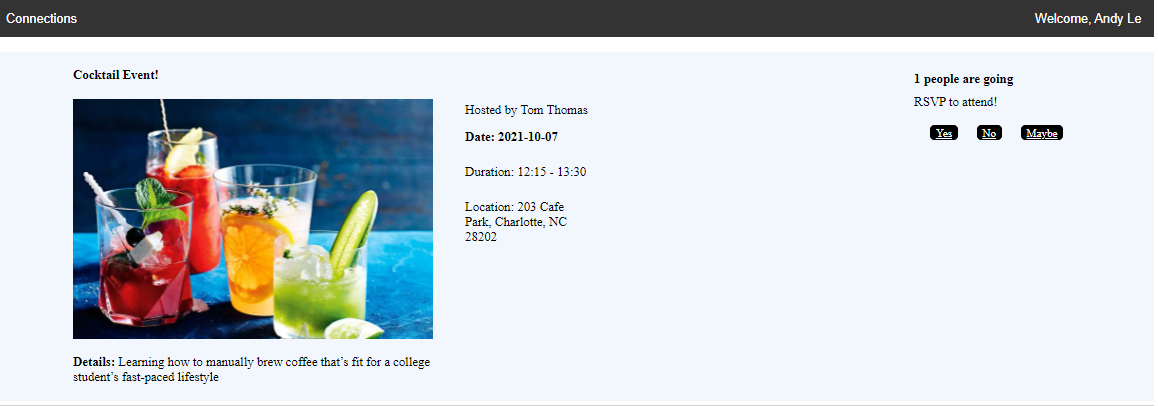
1. Updated Connection



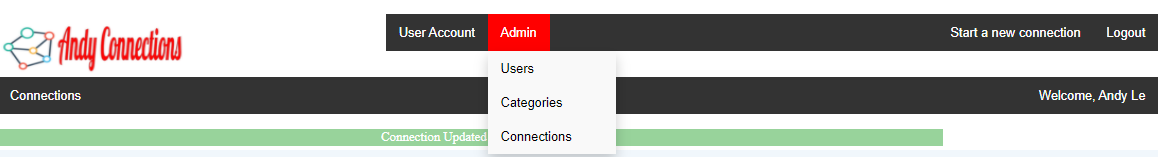
1. RSVP page



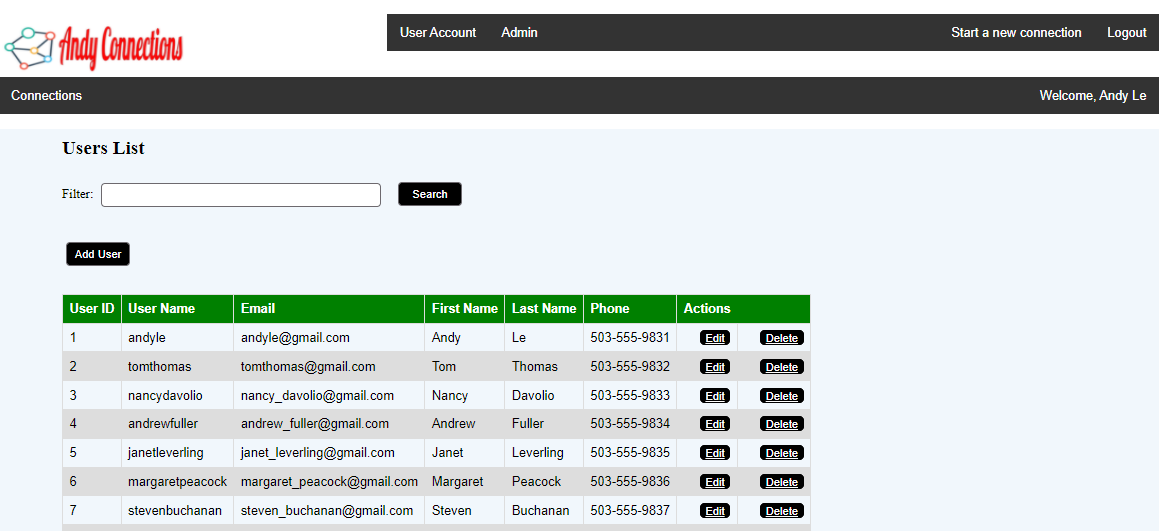
1. RSVP updated status



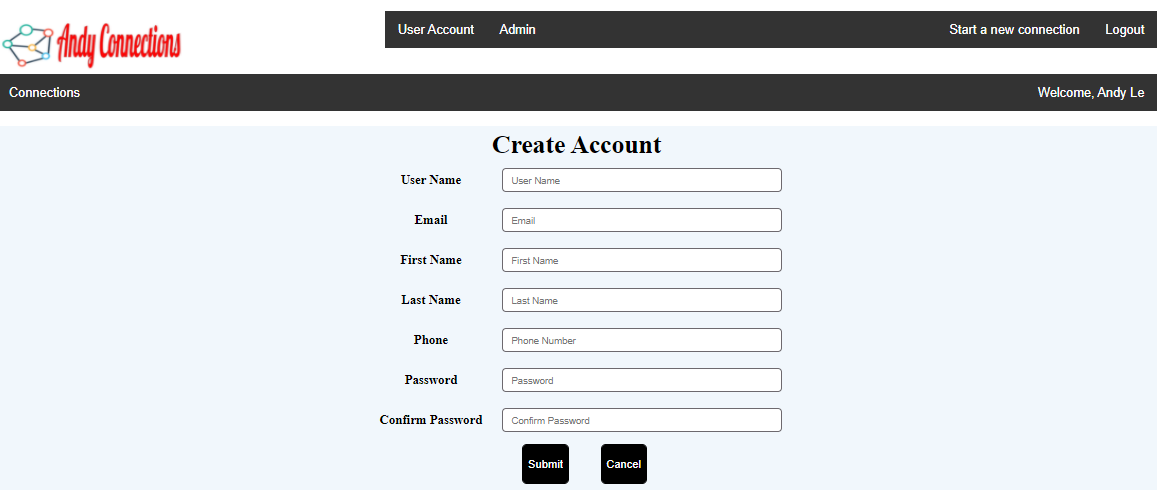
1. Admin Menu



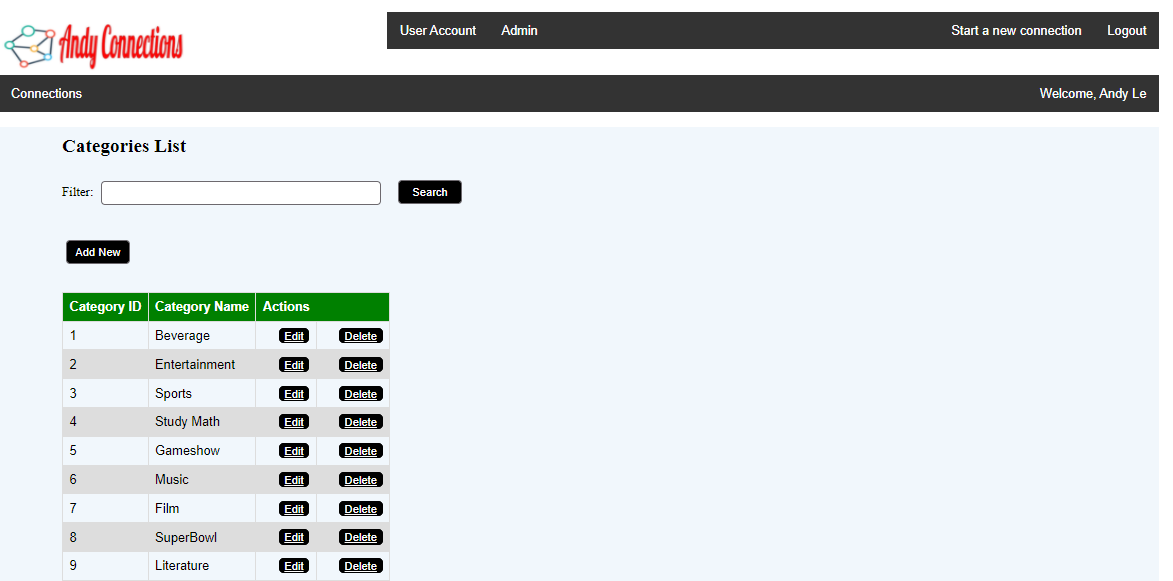
1. Amin - User Maintenance



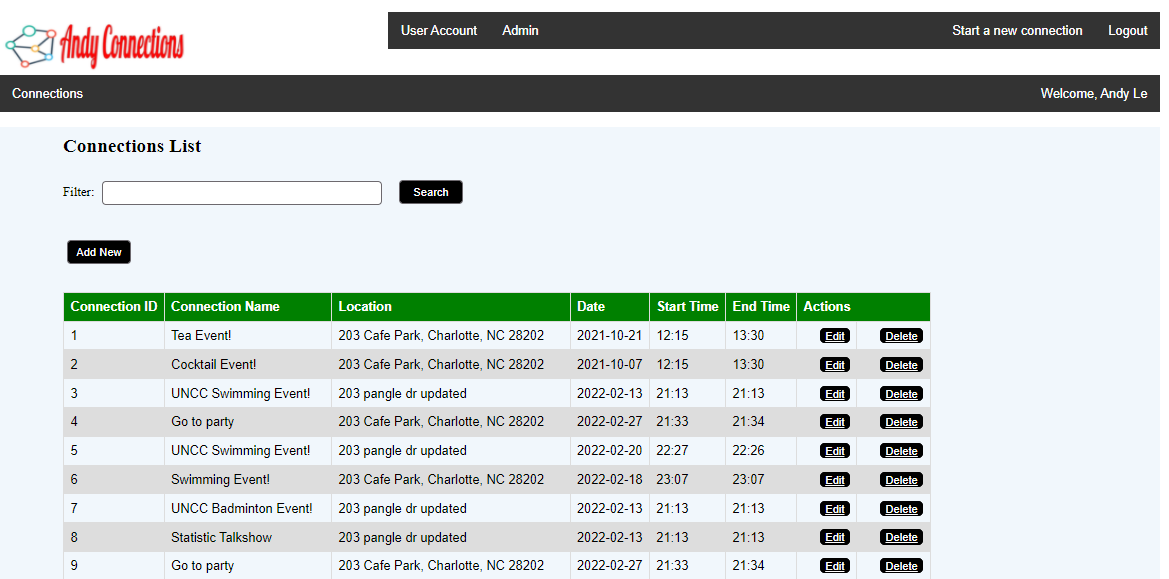
1. Admin - Create a User Account



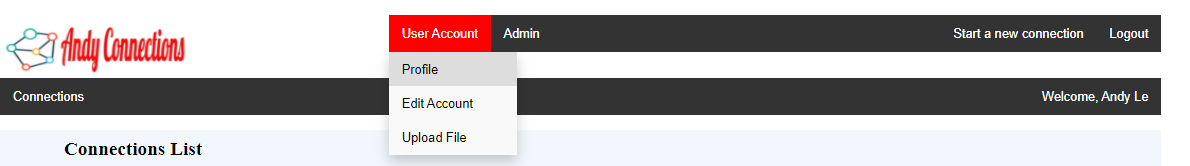
1. Admin - Category Maintenance



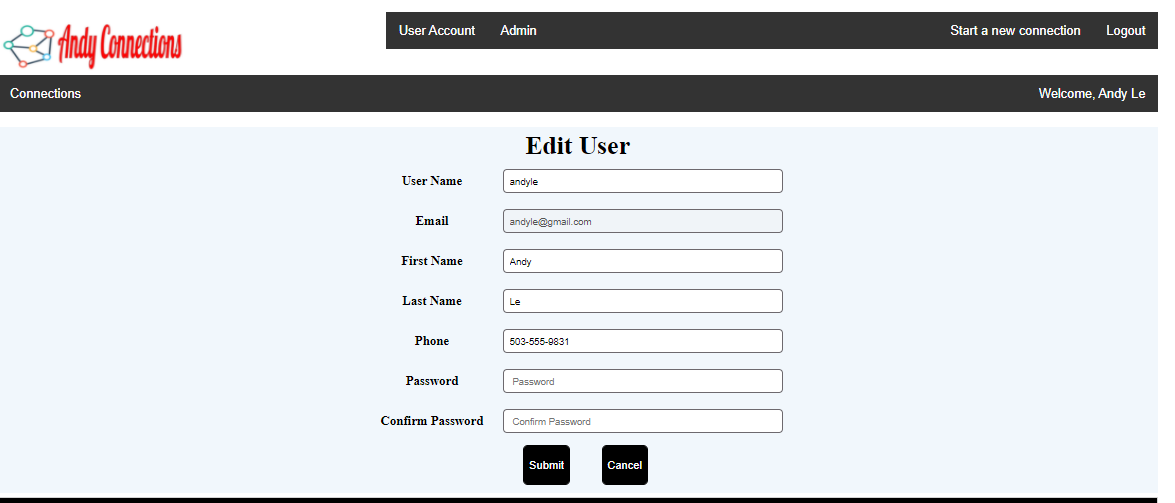
1. Admin - Connections Maintenance



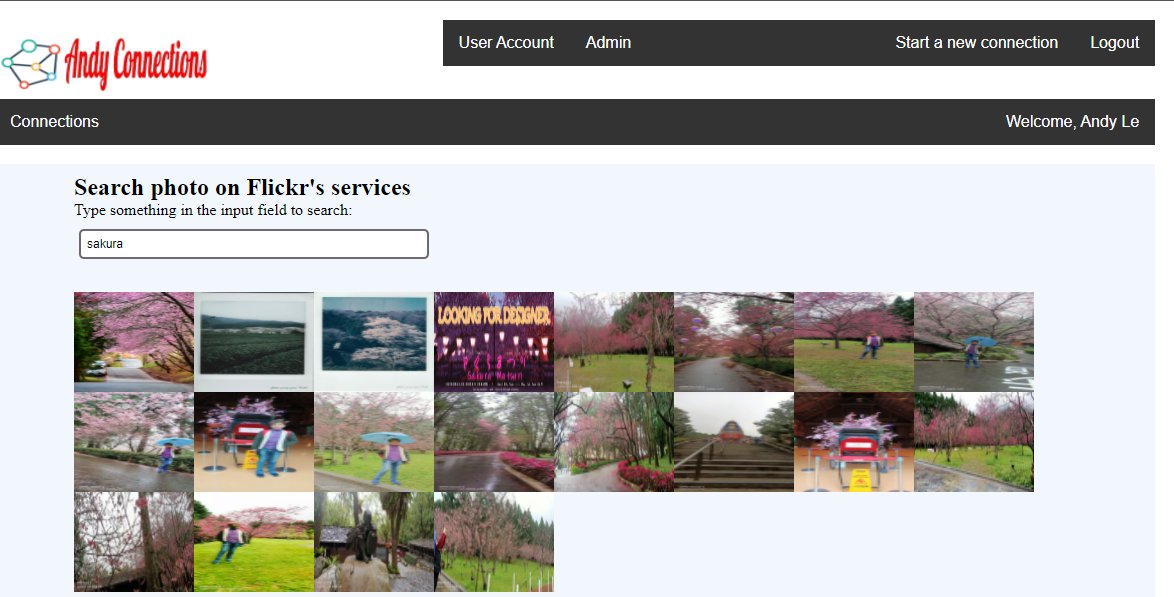
1. User Account Menu



1. User Account - Edit User

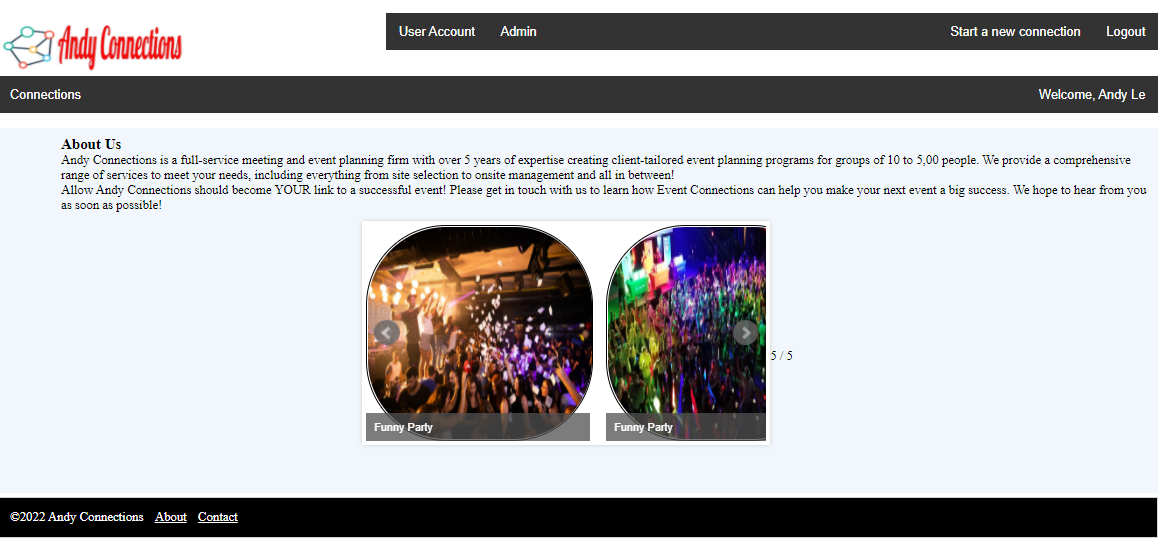


1. Search Photos using API functionality



Using the API search photos from photo services of Flickr. Do the call Ajax to return the data from Flickr's API and the return data in JSON format.

1. About Page



1. Contact Page

