

## Project Title: **Ticket Masterer**

### Project Summary:

Ticket Masterer is a platform that joins several online event ticket marketplaces' offerings into a unified shopping experience. Users can then search for events and find the best ticket deals across all marketplaces. Search results include the product name, description, price, images, and the marketplace link. Users will also be able to sort results by those fields.

When a user finds a ticket they like, they can add it to their personal wishlist, which organizes their selected products by marketplace for easy checkout. The platform will pull data from StubHub, ETickets, Sports Illustrated Tickets, Tickets.com, FeverUp, Vivid Seats, and MainLineTix. Because each of these datasets includes the date, time, and location of each event, determining which events are the same will be trivial.

### Project Description:

Ticket Masterer is an efficient event discovery platform to transform how users find and purchase event tickets across websites. The platform features a search interface that allows users to explore events and tickets. He'll be able to apply multiple filters like city, price, time, etc. It also offers a personal wishlist feature that allows users to save tickets they are interested in. The wishlist organizes these tickets by marketplace, facilitating a convenient and organized review process. Users will be able to perform CRUD operations on this wishlist.

The landing page of the project will prompt the user to register for the site. After registration, the user will be added to the "Users" table. After login, they can enter their search criteria into the search bar, which may include event name, location or date. The result displays a list of matching events, price, description and a link to the ticketing marketplace. They can refine their search results if needed. Then, they click on the event to view more detailed information, including seating options, ticket availability, total cost and an option to wishlist it.

*Additionally, the admin user will be able to manipulate the tables associated with events and tickets. After login, he'll be prompted to see the existing events available to other users. After clicking on the event, he'll be able to edit the ticket details as well. Also, the admin can also get a view of the number of users and their wishlisted events.*

Users will also be able to see the analytics dashboard featuring the trends and gain insights into ticket sales, pricing trends, and event popularity.

### Creative Component:

The creative component of our project will be data visualizations. Specifically, users will be able to see a heat map of seat prices per section for each event. Every venue is different, so section and seat data will be used to generate and fill in a mockup

of the venue. Additionally, we must pre-process and clean the data, and convert the json files (S3 buckets) to SQL tables.

#### Usefulness:

This will be useful as the user can find the best deals for event tickets by comparing prices for various seats at their chosen event across many marketplaces without having to navigate through different websites.

There are several platforms (e.g. EventBrite, Stubhub, tickets.com, etc) which offer similar functionalities for discovering, purchasing, and managing event tickets. While they each have their unique strengths, Ticket Masterer aims to differentiate itself by integrating data from multiple sources into a single, unified platform, providing users with a comprehensive and streamlined ticket-buying experience.

#### Realness:

This application will be real because it will use real event and ticket price data from several ticket marketplaces. We will get the data from [StubHub](#), [ETickets](#), [Sports Illustrated Tickets](#), [Tickets.com](#), [FeverUp](#), [Vivid Seats](#), and [MainLineTix](#) as provided for free on the AWS marketplace then downloaded as json files. In total, the datasets have a cardinality of around 40,000 and a degree of 19 with columns event\_url, event\_id, title, datetime\_local, name, address, city, state, country, postal\_code, scrape\_date, ticket\_id, ticket\_price, total\_price, fee, full\_section, section, row, and quantity common to all datasets.

#### Functionality:

The program will work by using a search bar. You will be able to search up an event, location, or date and you will be able to compare ticket prices for various seats across many ticket marketplaces. The wishlist will allow you to keep track of the tickets that interest you.

#### CRUD Operations that will be performed:

- **Create**
  1. **Create a new User:**

Users can register for a new account by providing their email address and any other required personal information.
  2. **Create a Wishlist:**

Users can add events they are interested in to their personal wishlist.
  3. **Create a new Event:**

*Admin can add events and details related to it for other users to see.*
  4. **Create ticket entries:**

*Admin can add ticket details associated with an event.*

- **Read**

1. **Read Search Results:**

Users can view detailed search results for events, including ticket prices, descriptions, images, and links to the ticketing marketplaces.

2. **View Event Details:**

Read detailed information about events.

3. **View Wishlist:**

Read wishlisted items.

- **Update**

1. **Update Account Information:**

Users can update their personal details.

2. **Update Wishlist:**

Update the wishlist information.

3. **Update Event Information:**

*Admin can update event details like time, date, location, etc.*

4. **Update Ticket Info:**

*Admin can update ticket prices, row, section, etc.*

- **Delete**

1. **Delete from Wishlist:**

Remove events from the wishlist.

2. **Delete Events:**

*Admin can delete an event after the event was concluded.*

Advanced Database features:

1. **Stored Procedure:**

Run a stored procedure to display recommended events to the user based on his demographics. These events will be displayed to the user by default. This will take info of other users in the same location and access their wishlisted tickets and events.

2. **Constraints:**

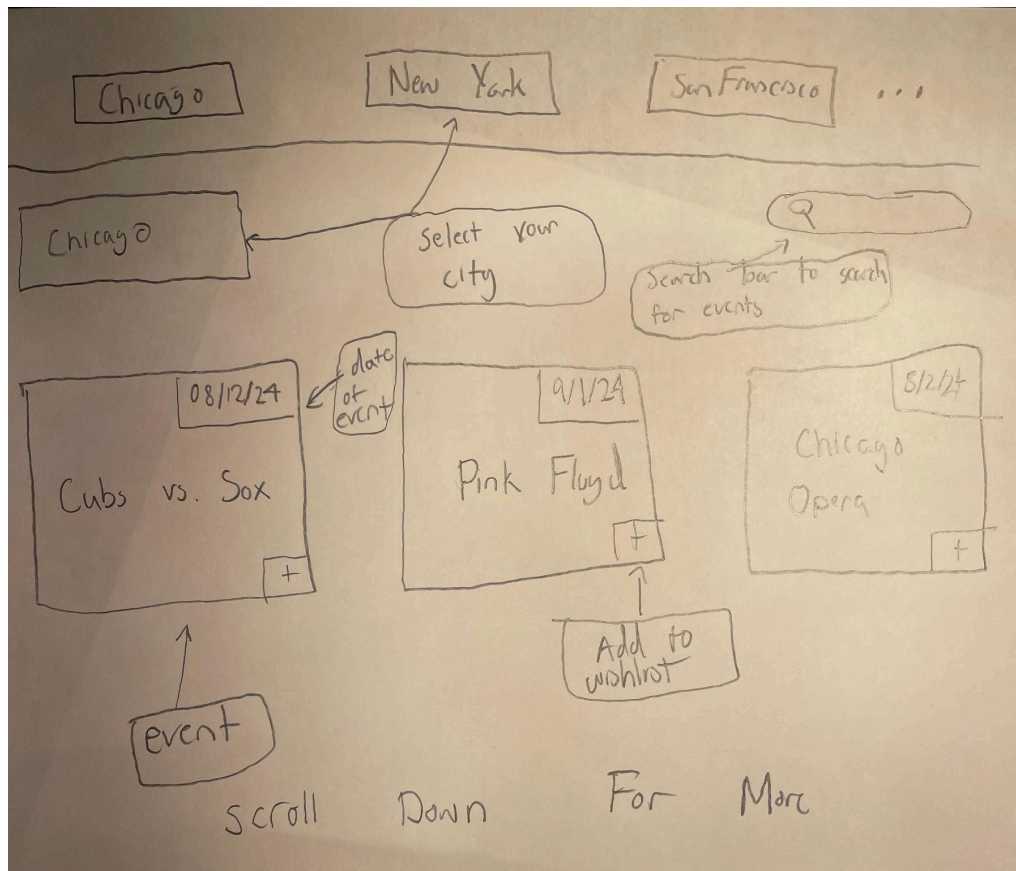
We will apply constraints for email format validation. Also if a user is deleted across the database, his wishlisted items will also be deleted.

Project Work Distribution:

- Mathew Ishaq will work on frontend development by building the wishlist feature.

- Nathaniel Dyer will work on back end development, specifically on integrating the many datasets into one and helping implement the creative component.
- Kevin Cruz Lopez will work on backend development, implementing the search in the dataset.
- Ajay Rao will work on managing data models, building appropriate relationships between the tables. He'll create additional procedures for performing CRUD operations over the tables.

### Low UI Mockup:



# Wish List

Date of event

