GigCove – Brisbane live music gigs

Liam Edwards-Playne   
Queensland Academy for Science, Mathematics and Technology

Contents

[Introduction 2](#_Toc393399152)

[Body 2](#_Toc393399153)

[Post a gig 3](#_Toc393399154)

[Gig review 4](#_Toc393399155)

[Gig discovery 4](#_Toc393399156)

[Conclusion 5](#_Toc393399157)

# Introduction

My project is a web application dedicated to the sharing and discovery of live music gigs in Brisbane. I chose my project because I noted a problem of music gig discovery in my community – while there are plenty of talented musicians and some places that they play at from time to time, there is no centralized listing of music gigs and venues. This is disappointing for two main reasons:

1. music is a form of entertainment that can be enjoyed by anyone, and local talent should be showcased and promoted better
2. live music should be embraced just as much as digital music because it strengthens community rapport between people

The main way that live music gigs are discovered in Brisbane is through word-of-mouth, which includes social network events and venues publishing information on their websites. I believe that by aggregating this data in a single place and providing a platform for it to be discovered it would greatly increase the popularity of live music and benefit the artists and community a lot.

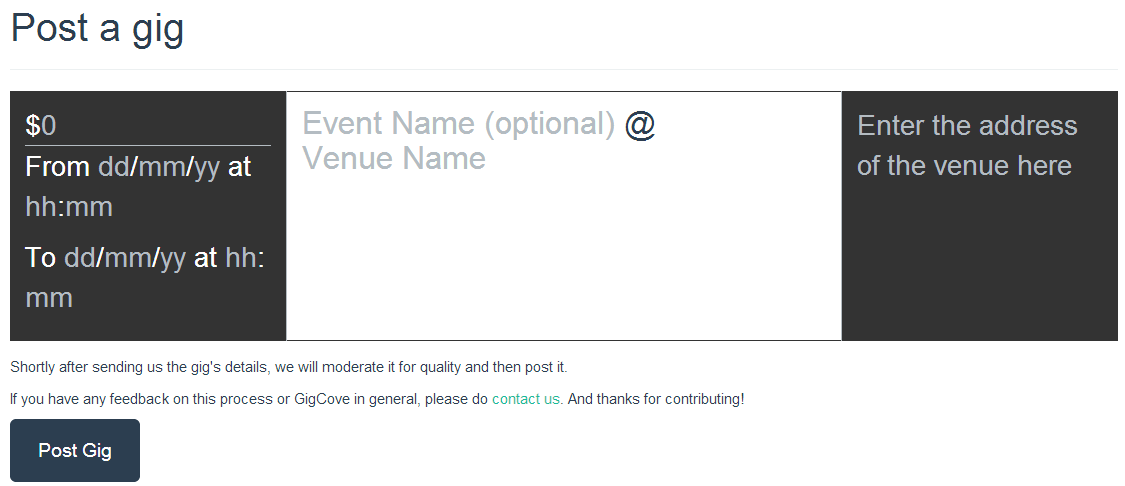
The web application is currently in the form of the minimum viable product, and consists of functionality for posting and discovering gigs. I anticipate that after I have more time that I will integrate the feedback into adding new features (these are discussed in the body).

The main problems I encountered when developing this were premature optimisation and lack of time. Originally this project began as a hobby when I was exploring Ruby on Rails, so I spent too much time exploring various aspects of optimising my server setup for scalability and security. Consequently, this combined with year 12 studies resulted in a very small amount of time. Nonetheless I have built an MVP suitable for easy improvement.

# Body

The web application is built using the Ruby on Rails web application framework. I use the Bootstrap 3 CSS framework with the LESS CSS language to style the web application responsively for desktop and mobile. The site features 3 principal areas: post a gig, the gig review menu and gig discovery.

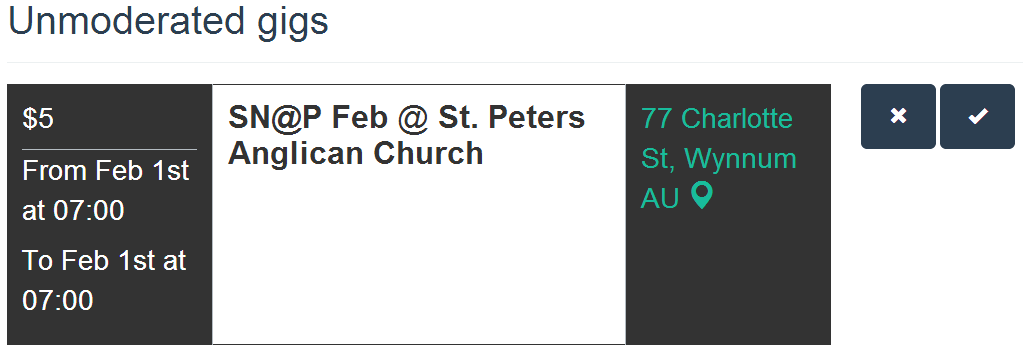
## Post a gig



Above is shown the form for posting a gig. The form is constructed using the contentEditable attribute of HTML5, which allows for seamless integration of text editing into typical HTML user interfaces. I took user experience very seriously when designing the project, as I knew it would have to be simple yet stylish for people to like using it. Using contentEditable I was able to construct the gig submission form in the exact same way that it would be displayed on the listings.

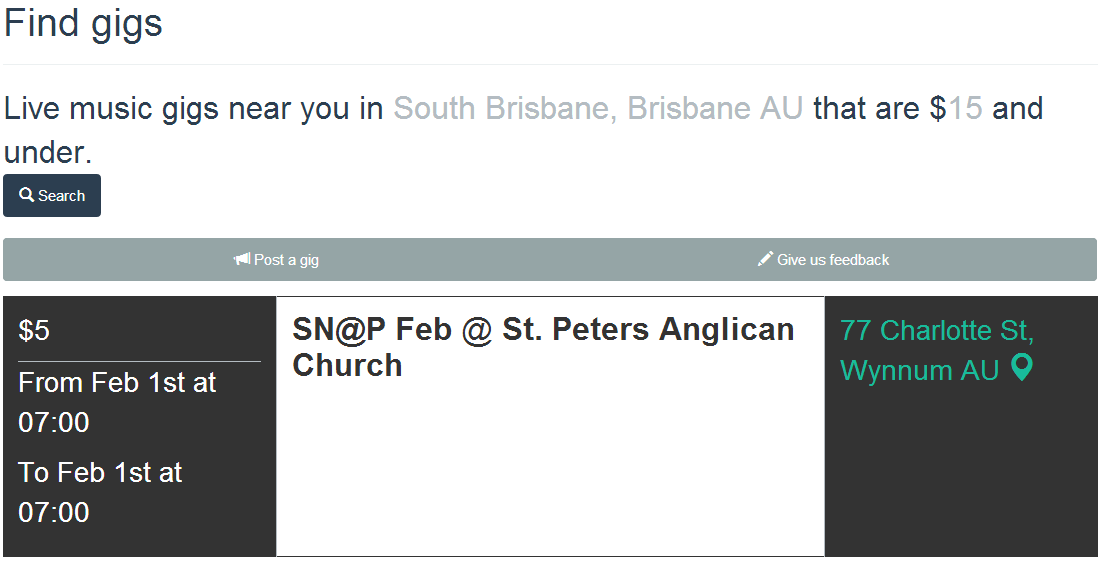
When data is inputted and submitted, the web application parses it and stores it in a database. In particular, the address of the venue is geolocated using the free MapQuest service in conjunction with the geocoder gem and the latitude and longitude is logged (the reasoning for this is discussed in the gig discovery section). Other fields are stored accordingly in database tables, which facilitates easy lookup using search criteria.

## Gig review



The gig review page shows a list of unmoderated gigs, those that have been submitted to the website and not yet publicly displayed. The moderation process is very simple – a gig can either be approved or not, the former resulting in it becoming available in the gig discovery process. For now, only registered administrators (as is determined in the application code) can access this page, which is ensured via the Devise plugin.

## Gig discovery



The gig discovery process is the most important page of them all, because ultimately it is the reason most users would come to the website. Equal attention was paid to user experience and design on this page.

The first thing to notice is that, once again, a seamless experience is provided using contentEditable. Instead of a clunky search form, GigCove is built on a natural-looking interface. The first line of text is represented using natural language, with grey colour being used as a visual indicator for user input. However this is not enough for a seamless user experience – using the HTML5 geolocation API, the location field is automatically filled in (South Brisbane) and defaults are set for the money. Finally, it is to be noted that the gig is displayed in the same way as on the gig submission form.

When a user clicks search, AJAX is used to retrieve gigs relevant to the user’s search parameters without refreshing the page. The ranking of gigs is based on the closest gigs first, an intentional decision to foster gigs in the local community. One technical design decision considered in this process was that of calculating the distance between the user and the gig. As the location would be different for every user, this would entail a new lookup for every search. For a free service like GigCove, this wouldn’t be sustainable profit-wise. So instead of doing this, the user sends their location (latitude+longitude) and the app uses a mathematical formula called the [Haversine Formula](http://en.wikipedia.org/wiki/Haversine_formula) to compute the distance computationally cheaply and very quickly (as opposed to using a third-party service to perform the distance check).

Finally there is also a menu bar between the search form and the results which contains two buttons – one for giving feedback and another for submitting gigs. Research has shown that providing users with a call to action stirs them, so it is intended that providing these clearly marked buttons will facilitate more feedback and gig submissions being received.

# Conclusion

When building this application I learnt the importance of planning out my time and not acting impulsively to fix some insignificant problems. Originally I felt it necessary to simply build the site to fix the problem, but it has largely served as a learning experience in building web apps too. Finally I’ve also learnt to understand the importance of good user experience and design.

## Features planned

My plan for improvement of this product includes three initiatives:

1. Improving the gig data – adding fields for the genres, bands/musicians playing, age restrictions, a short description and/or poster and finally a link to the source.
2. Improving the gig submission process – automatically fetching data about new gigs from known sources on venue’s websites.
3. Fostering a user discussion system – including a forum/discussion platform to discuss/promote gigs, rating system for venues