FULL STACK APPLICATION TO CONNECT MUSICIANS IN IRELAND

Final Year Project Semester 1

(20096595) Dean Doyle

TABLE OF CONTENTS

Introduction:

I will be designing and building an application to connect musicians. There are many opportunities for working musicians in Ireland and this application will aim to bridge the gap between musicians seeking performance opportunities and acts in need of additional members to cover gigs. By streamlining the process of making these connections there is the potential to enhance professional opportunities for musicians and support the growth of the live music scene in Ireland.

Background:

My own personal background in the music industry in Ireland led me to decide to build this solution for working musicians. I started out playing with bands about 15 years ago, at this stage of my career I would play with one band at a time and was sufficiently busy. After a few years of this I decided to make it more of a full-time job and it was at this point I realised that working with one act would not supply enough work to sustain a full-time job as a sole trader. Through the years up to this point I had built up a good network of venues with which I had a good relationship with and therefore noticed I had the opportunity for more work but not enough availability of the other musicians in the single band I was in to fill these gigs. This led to a loss of revenue as I was having to turn down gig offers from venues. This in turn led me to try becoming more flexible and able to stand in with any acts that needed someone to fill in for a gig, this combined with my connections with the venues allowed me to book acts and insert myself into them as an extra member with no added work to the current act as I would learn their material. This benefited the acts as they were getting more work, and it benefited me as I had more available musicians to choose from to fill any gigs I was being offered.

Current Solutions:

After doing this I then noticed there were groups of other people who were doing something similar but on a larger scale. Specifically, they would only stand in with acts and were not a regular member of any one act. There are various groups on Facebook such as “Need A Dep Ireland (For Musicians)” which allow musicians to post their availability and profile themselves to get more work. I noticed a trend in the posts which seemed that they were consistently posting up the same kind of format of post for example:

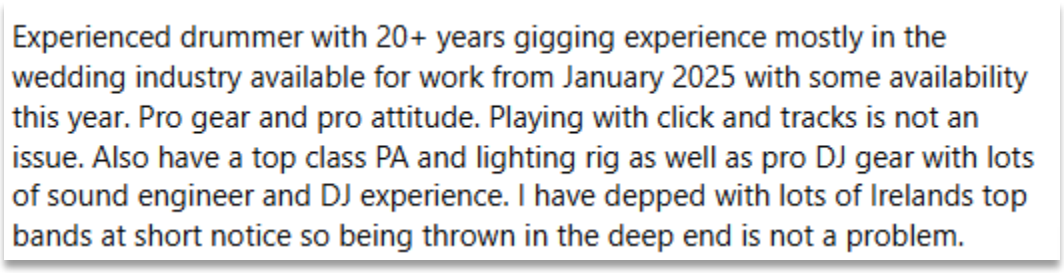

Figure 1: Need A Dep Ireland post – Drummer looking for work.

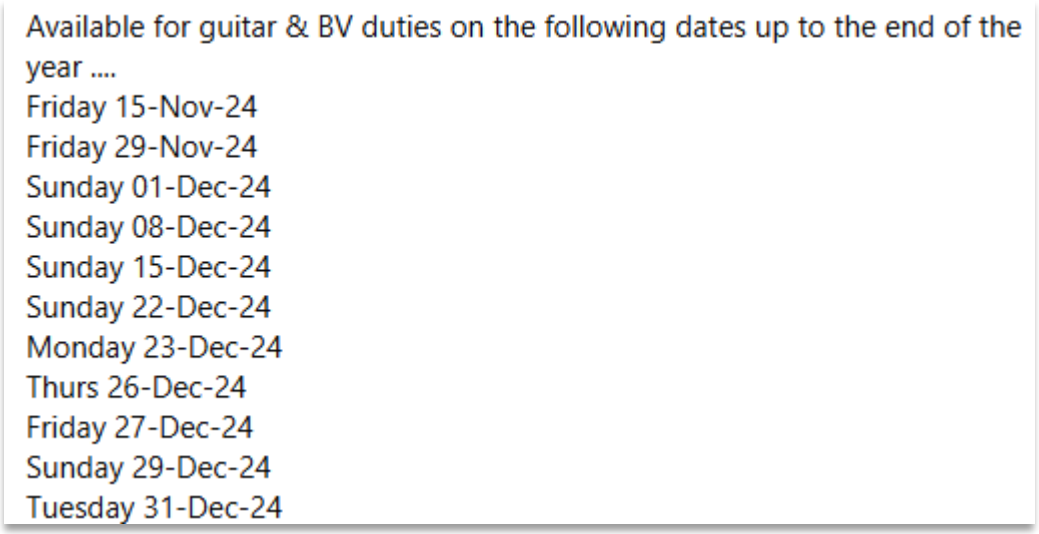

Figure 2: Need A Dep Ireland – guitarist with backing vocals posting availability.

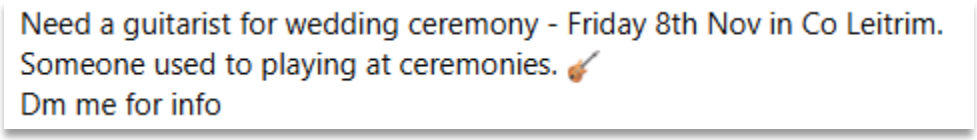

Figure 3: Need A Dep Ireland – Request for guitarist for wedding ceremony.

From the above screenshots we can see some various posts from musicians looking for work but also someone requesting and available musician to fill in a position needed for a wedding ceremony.

Another group is Dep. Musicians in Ireland – Wanted & Available

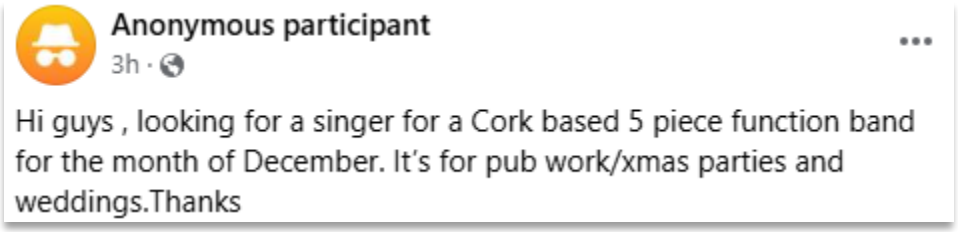

Figure 4: Dep. Musicians in Ireland – Wanted & Available – Request for a band.

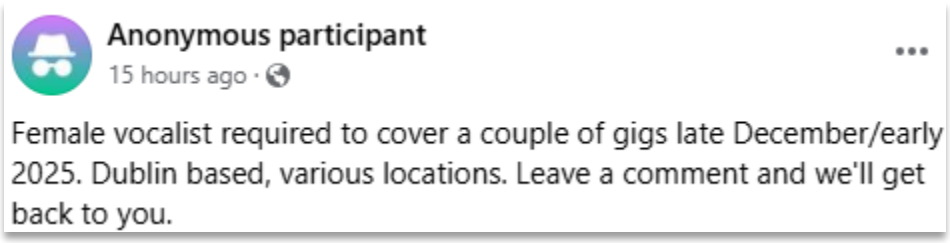

Figure 5: Dep. Musicians in Ireland – Wanted & Available – Female Vocalist request.

We can see a pattern forming across the groups, but they are still fragmented away from each other.

Dublin Studio Hub is a company which offers a service to musicians by providing a studio environment for recording needs but also offers available session musicians to play their instrument on your track.

"Tracks from top shelf session musicians are our speciality. Send us a mix of your track , minus the instrument you would like us to record, plus any musical brief or idea or sketch for the part, and we will work with you to get the exact sound you want, recording in our top shelf Dublin Studios - and we'll send you back a professionally recorded instrument track, all ensuring you are 100% satisfied." (Dublin Studio Hub, n.d.)

This is more of a studio-based solution for recording artists and not surrounding the gig market in Ireland.

Generally, there is not a consolidated solution for gigging musicians to connect and increase work availability in Ireland. The solutions available are fragmented across various private groups on social media.

User Stories

Musician:

- As a musician, I would like to be able to create a profile and showcase my skills with the idea of creating more work for myself.
- I would like to be able to apply for gigs as they become available within a set working area and relative to my instrument.
- I would like to have a way of keeping track of upcoming work I have accepted.
- I would like all correspondence to take place in the application.

Act:

- As an act, I would like to be able to post details of an upcoming gig where I am stuck for a position within the band due to one of the members being unavailable.
- I want to be able to search through available musicians in my area and have a way of decerning there ability and suitability for a gig.
- I would like to be able to keep track of upcoming gigs and which I have filled the positions I need.
- I would like to be able to keep all communication in one place so I can better keep track of discussions and avoid issues with bookings.
- I would like to utilise a rating system to lend credibility to a musician's reliability.

Objectives

- Design and build and application to connect artists with other artists looking to increase work.
- Consolidate and improve the currently existing solutions.
- Provide an intuitive platform for managing gigs and increasing opportunities for gigging musicians.
- Support the growth of the live music scene in Ireland.

Proposed System Description

The application I propose to create as solution to the issue will focus on consolidating the fragmented way these interactions between musicians in the industry are already taking place. Users will be able to create profiles and from these fit into two categories "Looking for Musician" and "Looking for Gig" at a base level.

<u>Looking for Musician:</u>

This represents a user who may be part of an existing act and is need of a musician to fill in a position for an upcoming gig. This user will be able to create a profile on the app and put up a gig opportunity post. The user will fill in some information about the gig such as the location and date and then also what position is needed to be filled. This will then be posted and made available for application by users who are in the "Looking for Gig" category. When an application is made by a musician a connection is made between the gig poster and the musician looking for work where further discussion can take place. In the scenario where the gig is accepted the gig is logged as "accepted" and after the date stated on the listing the user that posted the gig is prompted to give the user who accepted the gig a rating based on some previously outlined criteria such as "musicianship", "punctuality", "quality of gear". These ratings then help the user who took the gig to get further work.

<u>Looking for Gig:</u>

This represents a user who is possibly not part of an act that is available for work. This user will set up a profile containing basic information alongside what instrument they play and any more relevant information. The user will be able to set their availability and also the range/area they are willing to travel for a gig. When a gig is posted that is in the range of travel, matches their

instrument and is on a date in which they are set as available the user will get a notification, prompting the user about the details of the gig. If the user feels like they would like to accept the gig they are then connected with the user that posted the gig. After discussion has taken place in the case of where the user has decided to take gig, it is added to their calendar and after the date proposed by the gig listing, they will receive a rating from the user who listed the job.

Methodology

I plan to use a structured, iterative approach that follows modern software engineering practices. The following are some key steps in the product lifecycle of the application.

1. **Understanding the requirements**

   Initially I focused on understanding the need of musicians and acts that are working in Ireland. At first, I pulled on my own personal experience and had discussions with fellow musicians to see what solutions they would find useful. After this time was spent researching existing solutions which I discussed above, and all of this gave me a good scope of what is available and how I could create a centralised solution to the issue.

2. **Design Phase**

   This phase involved considering the users usage of the application, having identified the personas of the users and categorising them into two distinct groups I can then figure out the journey a user will take through interaction with the app.

   The key stages of user interaction are divided into:

   Onboarding: Sign up and profile creation.

   Exploration: Searching for jobs or musicians.

   Engagement: Posting or applying for gigs.

   Fulfilment: Completing a gig and managing feedback.

   By focusing on this user journey, I will be able to identify the needs of the individual user groups, compare them against the existing solution and my own industry experience to optimise the user experience.

3. **Development**

   This stage will focus on implementing the functionality based on the insights from the design phase. The backend will manage user action and data storage and the chat system, while the frontend will offer and intuitive user interface. Communication between the backend and front end will be through well defined APIs prioritising modern standards for performance and maintainability. The application will be hosted on AWS and utilise cloud services to maintain reliability and scalability. I will also adhere to best practises for testing and continuous integration to ensure a robust final product.

Proposed System Architecture

Backend development:

I plan on using Go as a backend technology having gained some experience with it outside of college. I feel like it's high performance and built in concurrency with goroutines will support the chat system well and manage the API endpoints for user actions and gig postings. Alternatively, I will use node.js and express.js as a back up as we have covered a lot of this in college but ideally, I would like to try integrating some of my outside learning within this project. The back-end functionality will be hosted on an AWS EC2 instance.

Go, also known as Golang, is a programming language developed by Google. It is characterized by its simplicity and efficiency and is used to develop a wide variety of applications, from command line programs to distributed systems and high-scale web applications.

Being an efficient language, Go is suitable for ***building applications that need to handle a high volume of concurrent requests****. In addition, its design emphasizes safety and error prevention.* (My Task Panal Consulting , n.d.)

Frontend Development:

I will use React as a frontend technology this will allow for easily created self-contained UI components, and this will communicate seamlessly with the Go backend via REST APIs.

React.js is an open-source JavaScript library, crafted with precision by Facebook, that aims to simplify the intricate process of building interactive user interfaces. Imagine a user interface built with React as a collection of components, each responsible for outputting a small, reusable piece of HTML code. (Herbert, 2023)

Authentication

This will be handled by Amazon Cognito, this is an easily integrated user management tool for secure authentication that automatically scales and is especially robust when used with applications hosted on AWS infrastructure.

Amazon Cognito is an identity platform for web and mobile apps. It's a user directory, an authentication server, and an authorization service for OAuth 2.0 access tokens and AWS credentials. (Amazon, n.d.)

Database Management

Aurora will be used to handle the structured data such as user profiles and gig details, this also provides an autoscaling solution to manage high traffic.

Amazon Aurora (Aurora) is a fully managed relational database engine that's compatible with MySQL and PostgreSQL. You already know how MySQL and PostgreSQL combine the speed and reliability of high-end commercial databases with the simplicity and cost-effectiveness of open-source databases. (Amazon, n.d.).

DynamoDB will be used for the chat system as it a NoSQL database with low latency, its built-in flexibility and scaling will be more suitable for a high-volume application such as a chat system.

DynamoDB is known for its impressive read/write performance even at extreme scale, and it's backed by all of the benefits that come with the surrounding AWS ecosystem, including high availability, on-demand scaling, fault tolerance, multi-region replication, and tight integration with other AWS services like Lambda, S3, IAM, etc. (Archer, 2024)

File Upload

This will be handled by Amazon S3 Buckets. This reliable and scalable cloud storage solution can support user uploaded files like profile pictures, audio demos and videos.

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. (Amazon, n.d.)

CI/CD

Git hub actions will be used to automate builds, testing and deployment. This integrates AWS services and automating these aspects of the development process will reduce manual errors and ensure that code is tested and deployed with each update lining up with the modern production standards for ongoing development projects.

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline. You can create workflows that build and test every pull request to your repository or deploy merged pull requests to production. (GitHub, n.d.)

UX Design

The initial reasoning for creating this app was to consolidate existing fragmented solutions for musicians trying to connect with each other. To address this issue the UX design will focus on creating an intuitive and accessible platform tailored to the needs of users. By understanding the distinct personas, we can build up an interaction flow for a typical user based on the key interactions from earlier.

On Boarding:

Musician: Sign up -> Create Profile -> Add musician information(instrument) and Availability -> Upload media(performance demos).

Act: Sign up -> Create Profile -> set search preferences.

Exploration:

Musician: Search for Gigs -> Filter by location and date -> View gig details -> Apply.

Act: Post Gigs -> Add requirements and details -> Receive applications -> View musician profiles.

Engagement:

Connect musicians and acts via messaging system -> discuss gig details -> Confirm.

Fulfilment:

Musician: Receive rating -> Update profile.

Act: Rate Musician.

This detailed user usage flow is a comprehensive overview of the typical user interaction with the application. Below is a visual representation of each of the distinct user groups journey through the application.

Musician

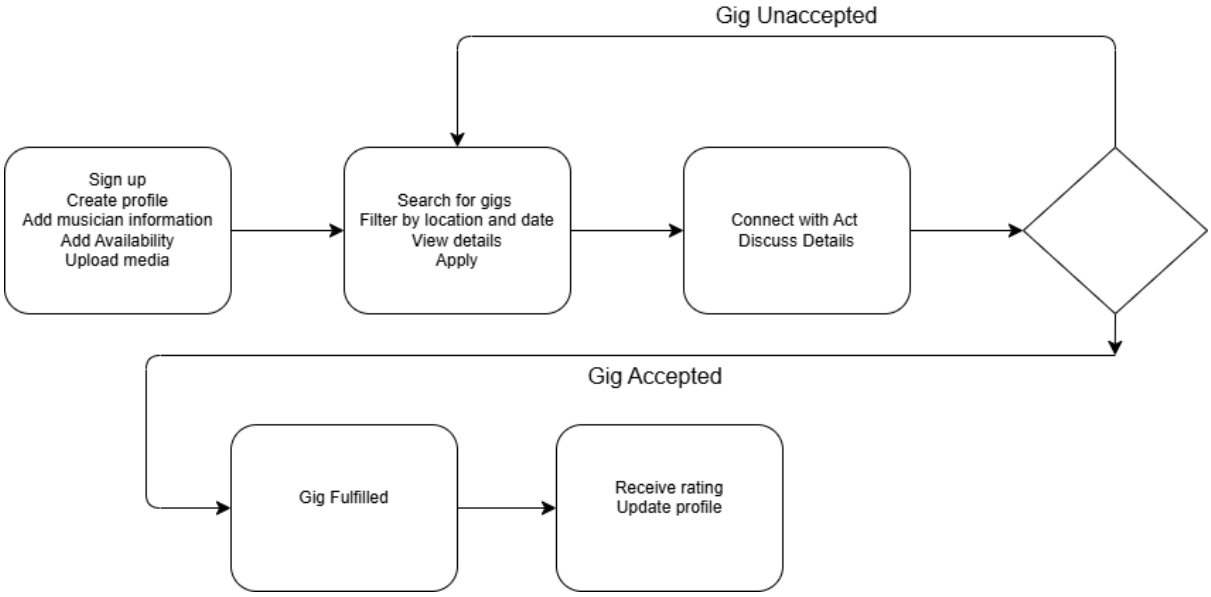

Figure 6: musician user journey

Act

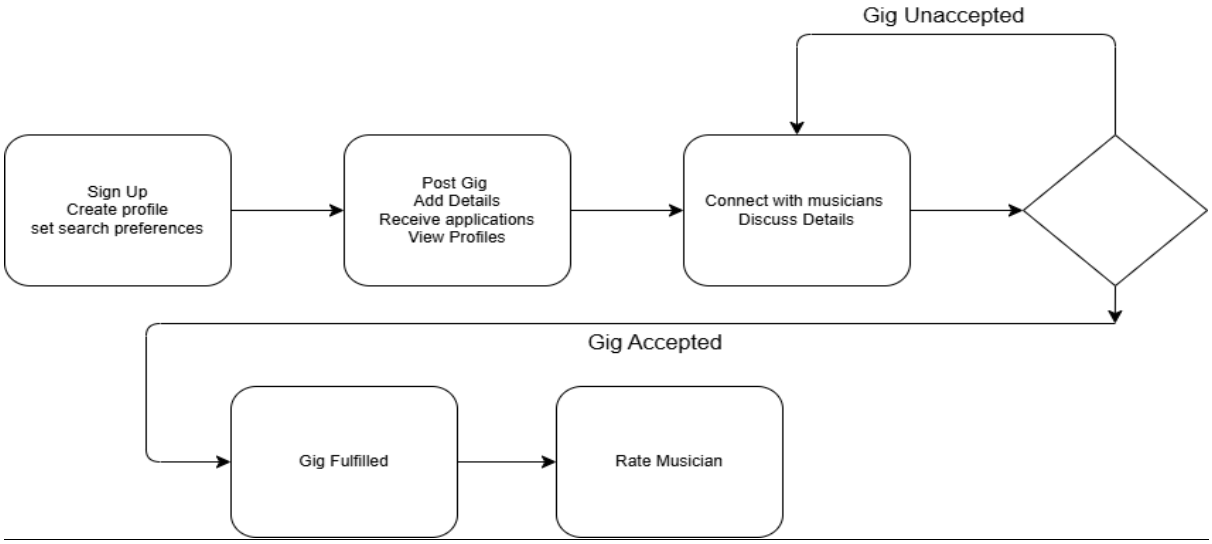

Figure 8: Act user journey.

Risk Assessment

This section will outline some potential risks that may impact the successful completion of the project and some strategies that might be implemented to mitigate these risks. As this application grows like any large project issues will arise as time goes on. By taking some time before production starts to think about some of the potential pitfalls that may occur, I can hopefully have some contingency plans in place to take a step through approach to solving any problems that arise.

Technical risks:

Integrating a lot of third-party services from providers such as AWS may introduce a level of risk due to limited exposure to these services in the past. While having used some of these services like DynamoDB and Cognito before I have never used other services such as Aurora. To mitigate this, I will utilise the extensive documentation AWS provides, online tutorials and take an iterative approach to development. By taking this approach, I should ensure the scope of troubleshooting will be smaller as features will be added as other features become operational.

Productivity risks:

By outlining a direct timeline and work schedule I will mitigate the risks involved with managing time and keep on track with the project's development. Implementing an AGILE methodology and working through feature-based sprints will allow for a clear distinction of progress. Having planned out the first half of the next semesters workload I have left myself room for flexibility, but the goal is to stay on track and add new features if the time allows. The main focus for weeks 1-6 is to create a base level functional application, that operates with the given features outlined earlier in this document. From the conception of this idea I always held that building outwards is crucial to a projects success and as features become available and operational the opportunity to add more will arise while maintaining that core functionality. I am aware of the risks to time management due to the workload of fourth year of college and will take steps outside of college to make sure that it takes precedence over everything else.

Users experience risks:

The user experience is core to the development direction of this project. The original concept being to consolidate existing solutions for this issue and therefore the application must be concise and easy to interact with. At various stages of production, I will be able to demo the application for people who work in the industry and use their reaction and feedback as a way to pivot the direction of the applications user experience to fit the needs of the users.

Security Risks:

I took the decision early on in the planning phase to create this application without using any real user data. The usual precautions will be in place and the considerations to be taken place as if this application were to be used with real data will be taken. Potential threats for data breaches and unauthorised user interactions will be handled with best software design practises and treated as though it were actual user data and compliance with standards such as GDPR will be adhered to but without the risk involved with handling that data.

Project Plan and Timeline

I will be using Trello as a means to plan out the timeline of this project. To this point I used it to outline the aspects of the research and design phase. As I move into the development phase I will create a more clearly defined set of steps through each part of development as they become functionally apparent.

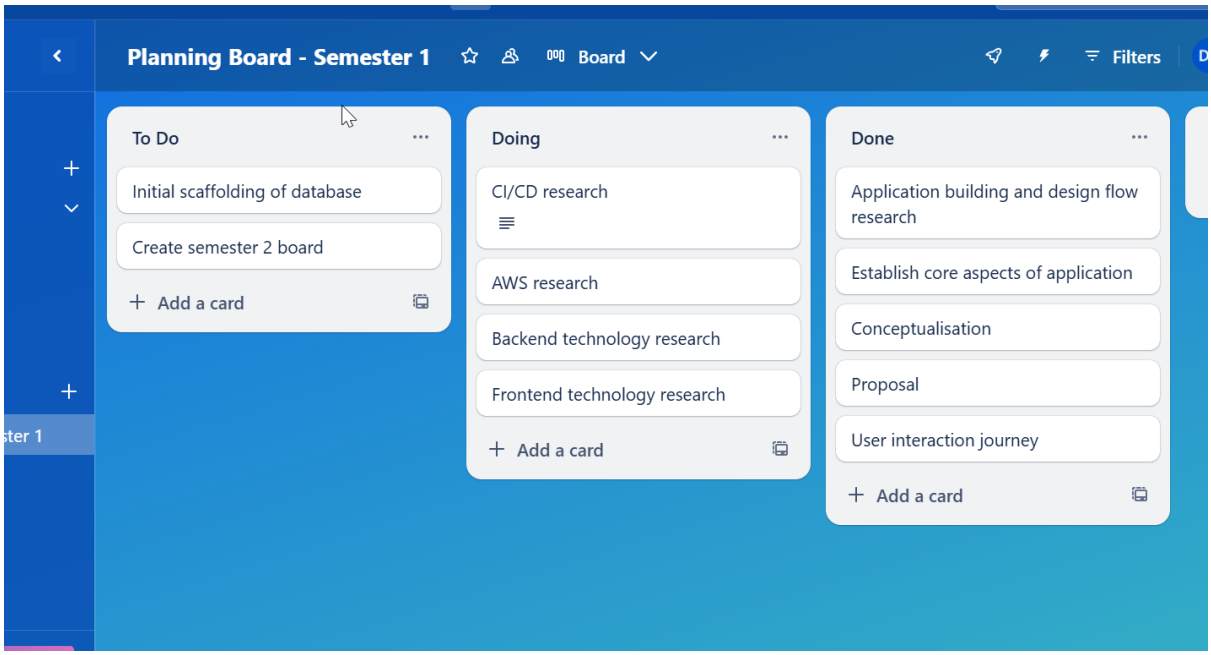

Figure 8: Research phase Trello board.

An example high level overview of the development phase will be as follows:

Week 1: Initial Set Up

CI/CD Pipeline Setup:

Configure GitHub Actions for automated builds and deployments.

Environment Setup:

Create and configure AWS EC2 instances for backend services.

Set up AWS S3 for file hosting.

Set up Aurora Serverless for database.

Week 2: Backend development

API development:

Implement REST APIs using Go for user profiles and gig postings.

Configure AWS Cognito for authentication.

Database development:

Design schema for Aurora Serverless for user profiles and gig details.

File upload:

Implement functionality for uploading profile images and performances using AWS S3 buckets.

Unit Testing:

Write unit tests for API end points.

Automate tests through GitHub Actions.

Week 3: Frontend Development

Frontend framework setup:

Initial React project setup.

Integration with CI/CD pipeline.

Develop core UI components:

Initial user functionality components:

User onboarding.

Profile management.

Gig posting and search interface.

Testing of backend/Frontend integration.

Ensuring components are seamlessly integrated.

Week 4: Integration of messaging service.

Set up and configuration of DynamoDB Database.

Developing flow from accepted gig to connecting musicians.

Week 5: Integration of rating system.

Addition of post gig completion functionality.

This initial overview covers the first half of the proposed timeline for the production of the application. This lays a good foundation for the base functionality of the application allowing for a level of flexibility for the lifecycle.

Conclusions

Undertaking a final year project such as this is an opportunity to bring together all of the knowledge gained over the last four years and merging that with something I am very passionate about. The Irish live music scene is vastly diverse and is very much driven by community. To build something that will support that while reenforcing the fundamental skills needed by a developer entering the workforce by planning and developing an application from conception to deployment is both challenging and rewarding. It provides a platform to showcase technical expertise and creativity while potentially having a positive impact on a community that has great meaning in my personal and professional life.

REFERENCES

Amazon, n.d. *What is Amazon Aurora*. [Online]
Available at: https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_AuroraOverview.html
[Accessed 28 12 2024].

Amazon, n.d. *What is Amazon Cognito*. [Online]
Available at: https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html
[Accessed 28 12 2024].

Amazon, n.d. *What is Amazon S3?*. [Online]
Available at: https://aws.amazon.com/s3/
[Accessed 28 12 2024].

Archer, C., 2024. *Top Use Cases for DynamoDB in 2024*. [Online]
Available at: https://www.tinybird.co/blog-posts/dynamodb-use-cases
[Accessed 28 12 2024].

Dublin Studio Hub, n.d. *Session Musicians*. [Online]
Available at: https://www.dublinstudiohub.com/session-musicians
[Accessed 07 11 2024].

GitHub, n.d. *Understanding GitHub Actions*. [Online]
Available at: https://docs.github.com/en/actions/about-github-actions/understanding-github-actions
[Accessed 28 12 2024].

Herbert, D., 2023. *What is React.js? Uses, Examples, & More*. [Online]
Available at: https://blog.hubspot.com/website/react-js
[Accessed 28 12 2024].

My Task Panal Consulting , n.d. *Go programming language: utilities, features and advantages*. [Online]
Available at: https://www.mytaskpanel.com/go-programming-language/#:~:text=It%20is%20characterized%20by%20its,high%20volume%20of%20concurrent%20requests.
[Accessed 28 12 2024].