Analysis Document

Name: Tony Jiang

Project: Music trivia game

Semester: 6

Version	**Date**	**Description**
0.1	7 March 24	Initial user stories
0.2	23 March 24	Initial non-functional requirements
0.3	25 March 24	Format the document structure and fix user story 10.
0.4	12 April 24	Modify document to analysis document.
0.5	31 Mei 24	Add Cloud

Contents

Context

This document represents the analysis document which contains the user stories, non-functional requirements, wireframe, and etc. The analysis document is for a music trivia game, it is a web-based game where players have 30 seconds to guess either the artist or the song name based on a 10-second preview of the song. This document outlines the project requirements, defines the target user, describes the user's interactions on the website and define the decision made to use a tool or a service to enhance or to test the project.

User story

Please note that these user stories are not final and may be updated over time until the final delivery week of the project.

There are currently two users, the player, and the admin on this system.

The user story has:

- **Estimation points (Est pts).** It ranges from 1 to 10 and is used to estimate the amount of time required to complete the user story. 1 rating is equal to 1 to 3 days and 10 rating is equal to 10 to 30 days. The point system for the minimum days is 1 x point amount. The maximum estimation points are 3 x point amount.

- **Priority points (Prior pts).** It starts from 1 and ends at 10. It is to rate which user story is important and must be done first. The higher the priority, the more important it is. 10 being the highest priority and 1 being the lowest priority.

- **Acceptance criteria**. It is like the definition of done. Certain stuff needs to be done in the acceptance criteria for the user story to be done/ complete.

User story 1

As a player, I want to select the genre of music and the year that I want to play, so that I can test my knowledge on that specific music.

Est pts: 2

Prior pts: 7

Acceptance Criteria

- I can pick which genres of music I want to play.
- I can pick which year I want to play.
- Start the game when the ready button is pressed.

User story 2

As a player, I want to have the option to choose how many rounds the session has, so that we can play longer or shorter rounds.

Est pts: 1

Prior pts: 4

Acceptance Criteria

- I can pick how many rounds I want to play till the round pick.
- I can see the rounds that I picked during the game session.

User story 3

As a player, I want to know which round I'm currently at, so that I know how many rounds are still left during the session.

Est pts: 1

Prior pts: 4

Acceptance Criteria

- It displays the current round.
- It displays how many rounds there are.

User story 4

As a player, I want to see the timer when I play the trivia game, so that I know how many times I have left before I lose the round.

Est pts: 2

Prior pts: 8

Acceptance Criteria

- The timer is displayed each round.
- The timer restarts every round.
- The timer should start when the song started to play.
- After the timer stop you won't be able to click on the answer.

User story 5

As a player, I want to see the correct answer when I pick the wrong answer, so that I know the right answer next time when the specific song plays.

Est pts: 3

Prior pts: 8

Acceptance Criteria

- I can see the correct answer by the color.
- I can see the wrong answers by the other color.
- It displays the answer I picked.

User story 6

As a player, I want to be able to invite my friends to play, so that I play with them and have fun.

Est pts: 5

Prior pts: 8

Acceptance Criteria

- I can see an invite button.
- I can invite my friends by they're name.
- I can see the empty player slot.
- I can see a ready button.
- I can see that my friend has join the session.
- There is a message box for the players invited to the session.

User story 7

As a player, I want to create a session, so that I can challenge players who join.

Est pts: 4

Prior pts: 8

Acceptance Criteria

- Random player can join the session.
- There is a ready button.
- The random people who join the session will be in the game.
- There is a message box for the players who joined the session.

User story 8

As a player, I want to be able to see the other players answers, so that I know what answer they have picked.

Est pts: 4

Prior pts: 7

Acceptance Criteria

- The other players answer is displayed.
- Each player is displayed differently.

User story 9

As a player, I want to see the total score of each player during the game session, so that I know who is winning.

Est pts: 3

Prior pts: 7

Acceptance Criteria

- I can see my total score each round.
- I can see the other players total scores each round.

User story 10

As a player, I want to see the total score of each player after the end of the whole round, so that I know who won this session.

Est pts: 3

Prior pts: 7

Acceptance Criteria

- Display the total scores of each player at the end.
- Display the total scores from high to low.
- Display a “Play again” button.
- You can play again after the score is displayed.

User story 11

As a player, I want to know the specific song name after the end of the whole round, so that I can go listen to that specific song.

Est pts: 2

Prior pts: 5

Acceptance Criteria

- I can see all the artist’s names and song names from each round after the end of the game session.

User story 12

As an admin, I want to add songs on the trivia game website, so that I can easily include the song for the game.

Est pts: 2

Prior pts: 9

Acceptance Criteria

- it’s required to add the genre and the year of the song.
- it's required to add the name of the song and the artist’s name.
- You can add the wrong answers.

User story 13

As an admin, I want to see all the added songs, so that I know which songs are already in the trivia game.

Est pts: 1

Prior pts: 9

Acceptance Criteria

- I can see all the songs by song names and artist named.
- I can filter by genre and by year.
- I can click on the song to see more details.

User story 14

As an admin, I want to be able to update a song, so that I can correct my mistakes.

Est pts: 1

Prior pts: 9

Acceptance Criteria

- I can update the gerne and the year of the song.
- I can update the name of the song and the artist’s name.
- I can update the wrong answers.

User story 15

As a player or an admin, I want to be able to login to the web app, so that I see my information.

Est pts: 1

Prior pts: 7

Acceptance Criteria

- I can see my profile page.
- I can edit, delete, and add some information on my profile page.
- I can see a logout button.

User story 16

As a player or an admin, I want to be able to logout of the web app, so that another person that uses my system doesn't have access to my account.

Est pts: 1

Prior pts: 7

Acceptance Criteria

- I can't see my profile page.
- I can still see a sign-up button.

Non-Functional requirements

Here are the define non-functional requirements for this project and their criteria.

- **Performance:** The game interface should respond to interaction during the game or outside the game within 1 second for 95% of the interaction.

- **Scalability:** The system should be able to handle at least ten thousand concurrent players during peak hours with minimal response issues.

- **Security:** The system should comply with relevant data protection and security standers like OWSAP.

- **Integration:** The testing framework should integrate seamlessly with continuous integration and delivery (CI/CD) pipelines, enabling automated testing as part of the software development lifecycle.

Wireframe

These are the initial wireframes of the website. They are in chronological order. I will also provide additional explanations on how you end up on a certain wireframe.

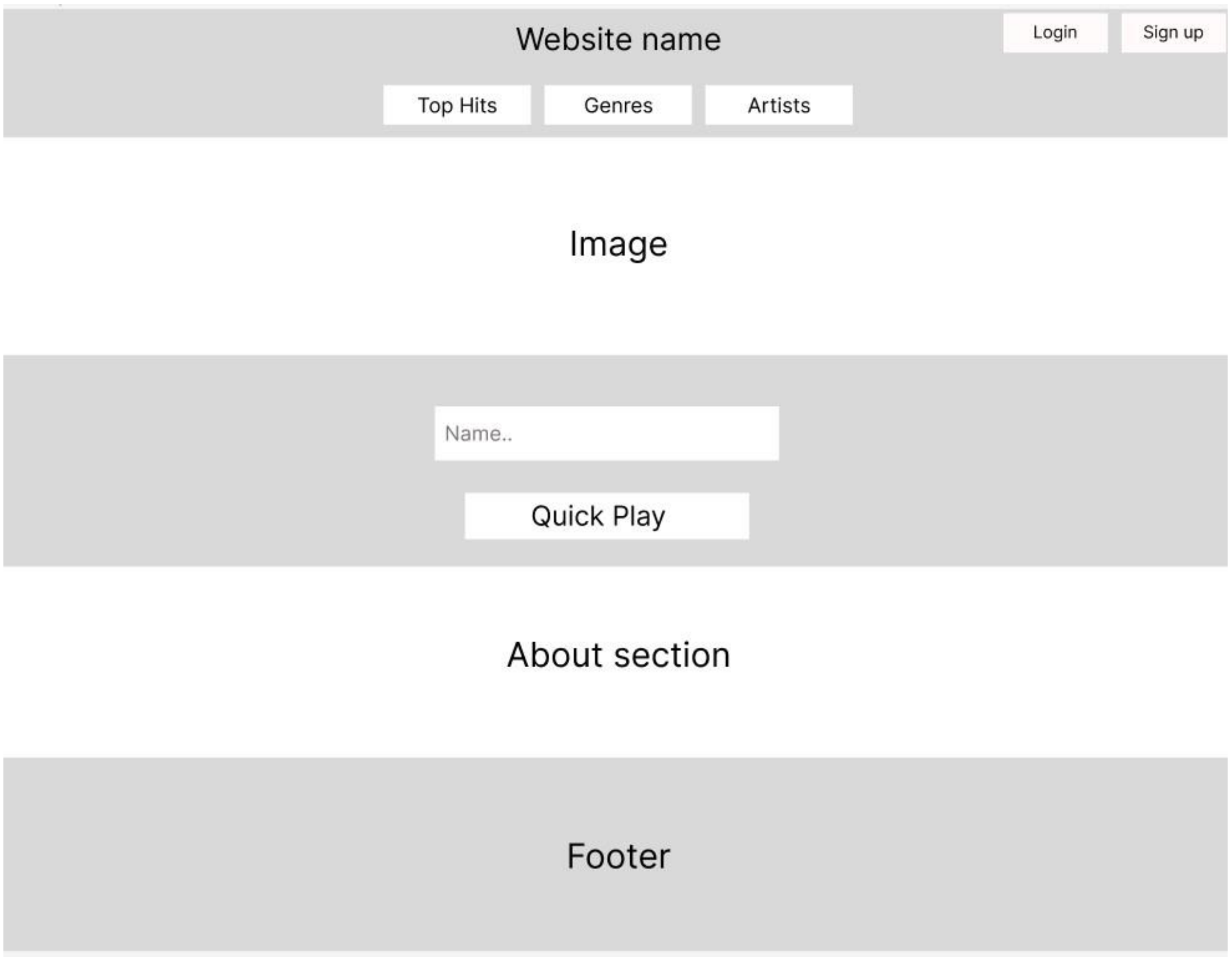


Figure 1: Home page.

This wireframe is the home page of the website.

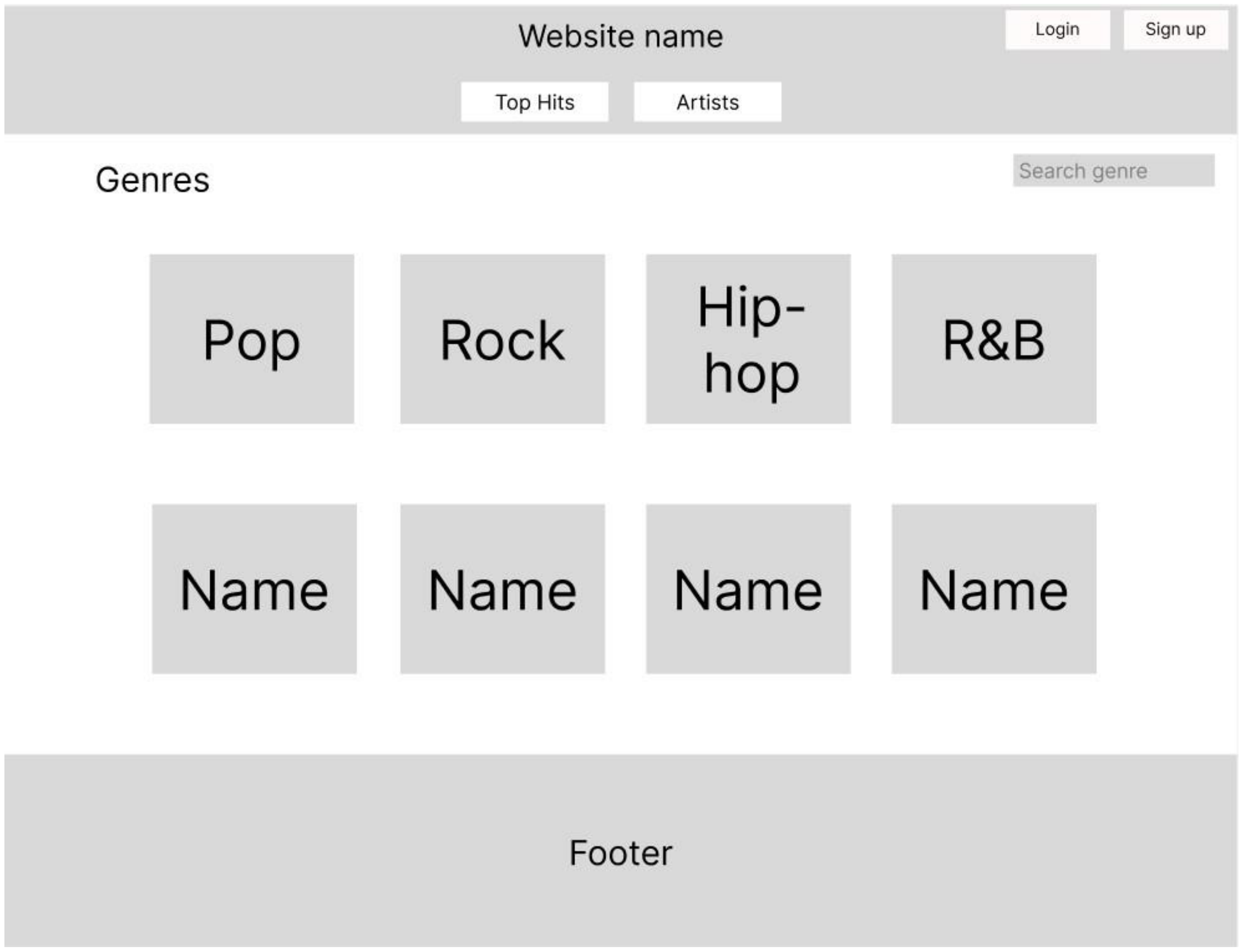


Figure 2: Genre page.

This wireframe is displayed when the "Genres" tab is clicked on. In this wireframe you would pick which genres you would want to play.

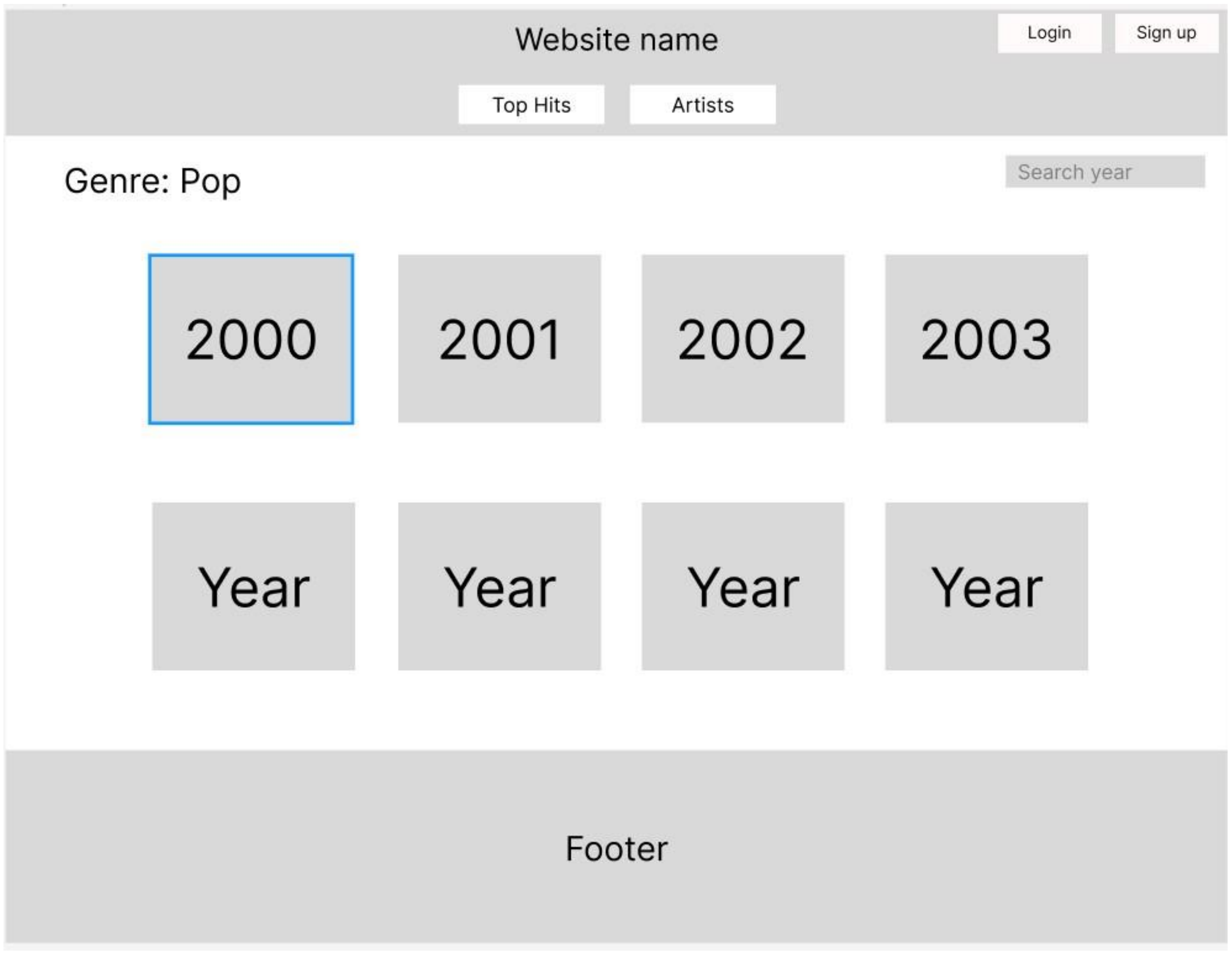


Figure 3: Year page.

This wireframe is displayed when you picked which genres you wanted to play. In this wireframe you would pick a year of the genre you wanted to play.

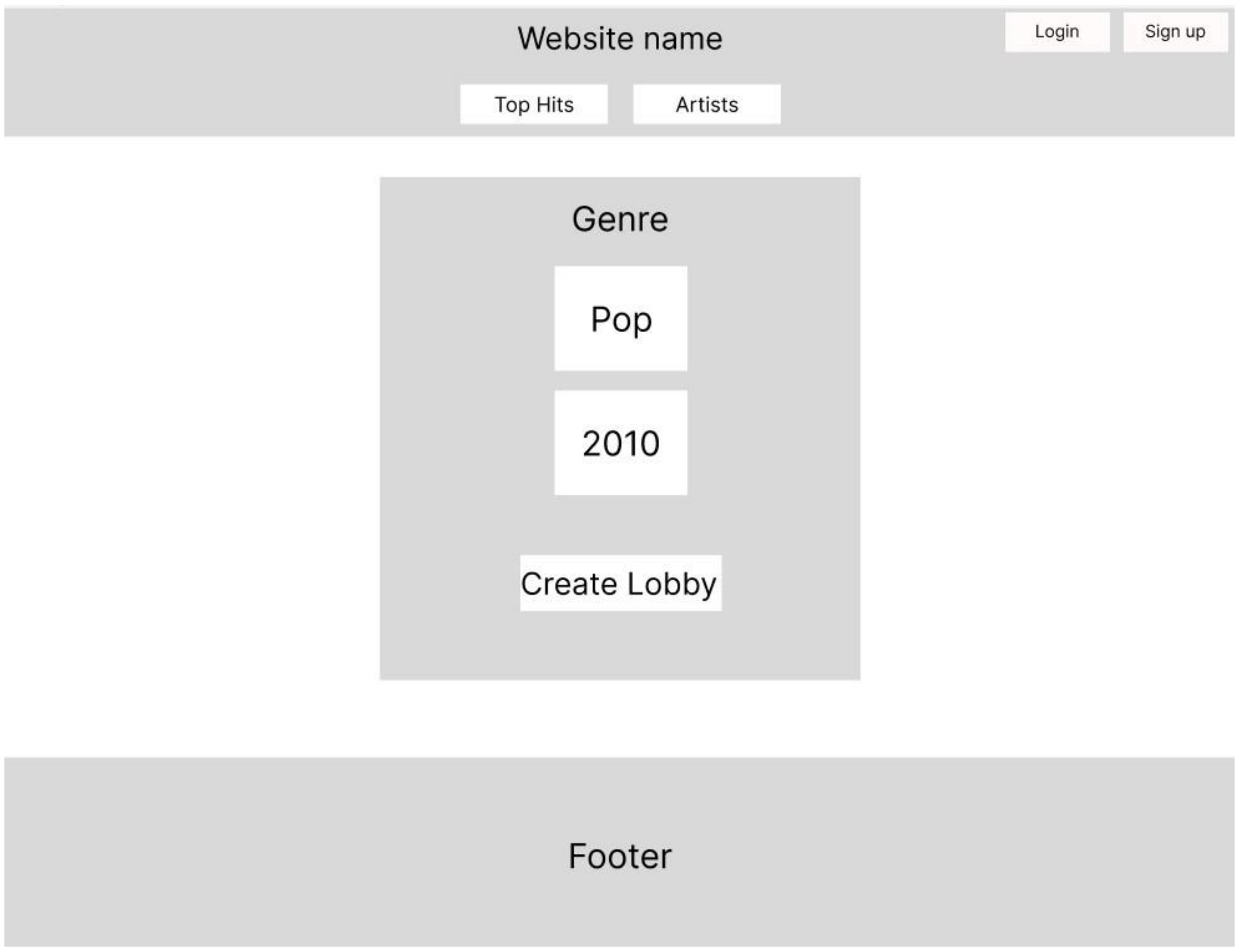


Figure 4: Initialize lobby page.

After picking the gerne and the year, this wireframe would display. In this wireframe it would display the genre you picked and the year and this is where you create the lobby to play the things you have picked.

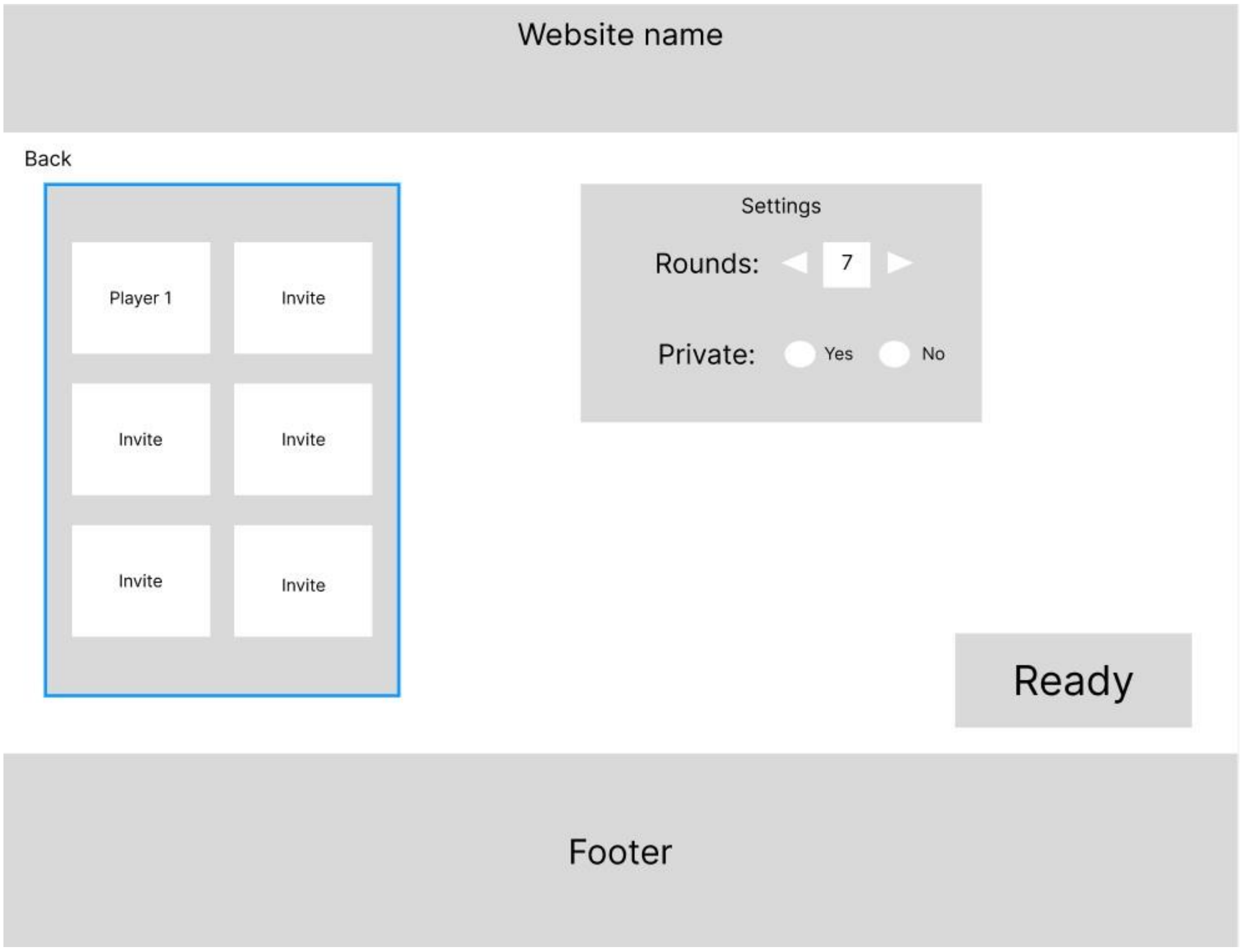


Figure 5: Create page.

This page is displayed after you clicked on the “Create Lobby”. In this wireframe you can invite people to play with you or play by yourself. You can also setup the game to go how many rounds and to make the lobby private or public.

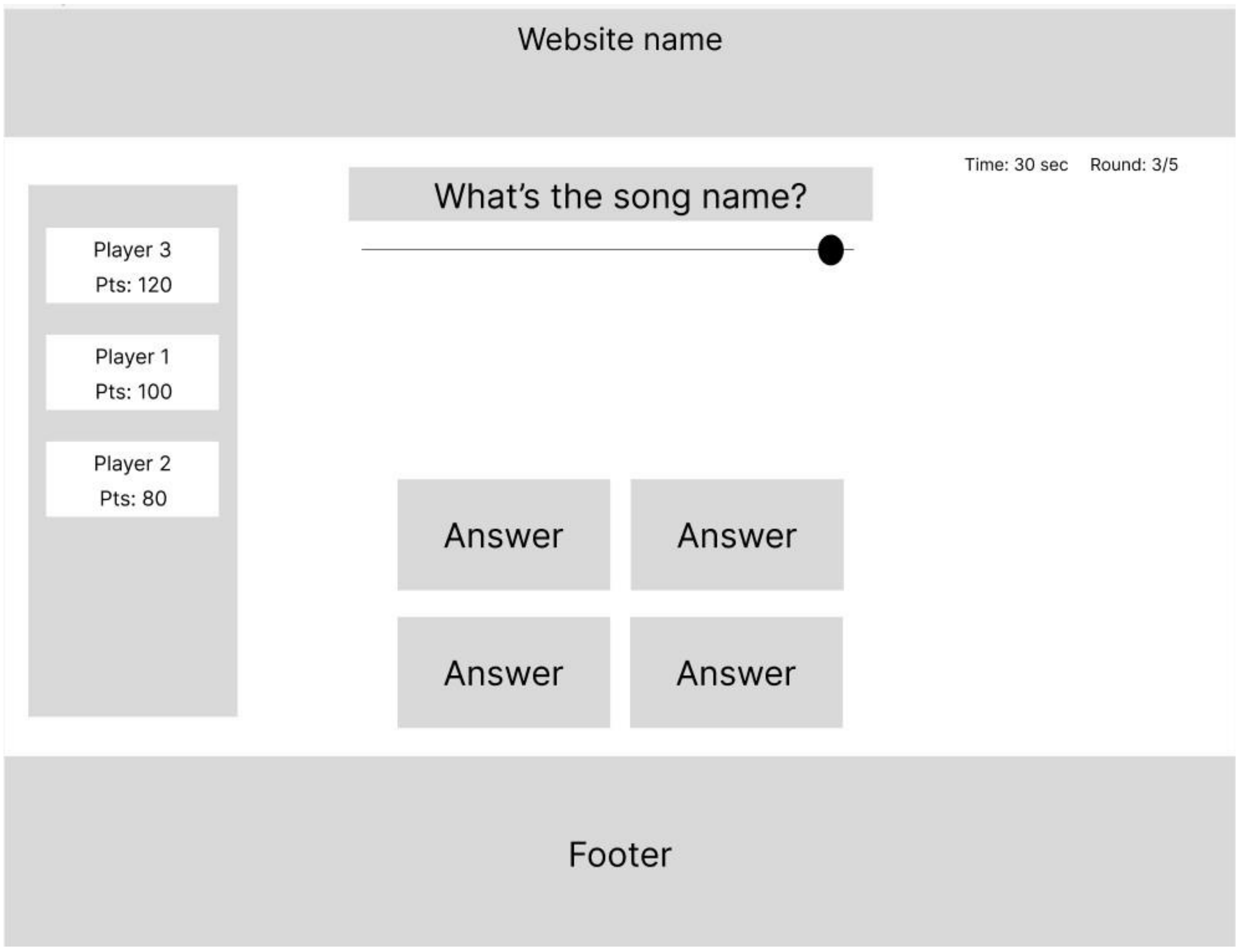


Figure 6: Game page.

This part of the wireframe is after you've clicked on the "Ready" button to start the game. This wireframe displayed how the game would look like.

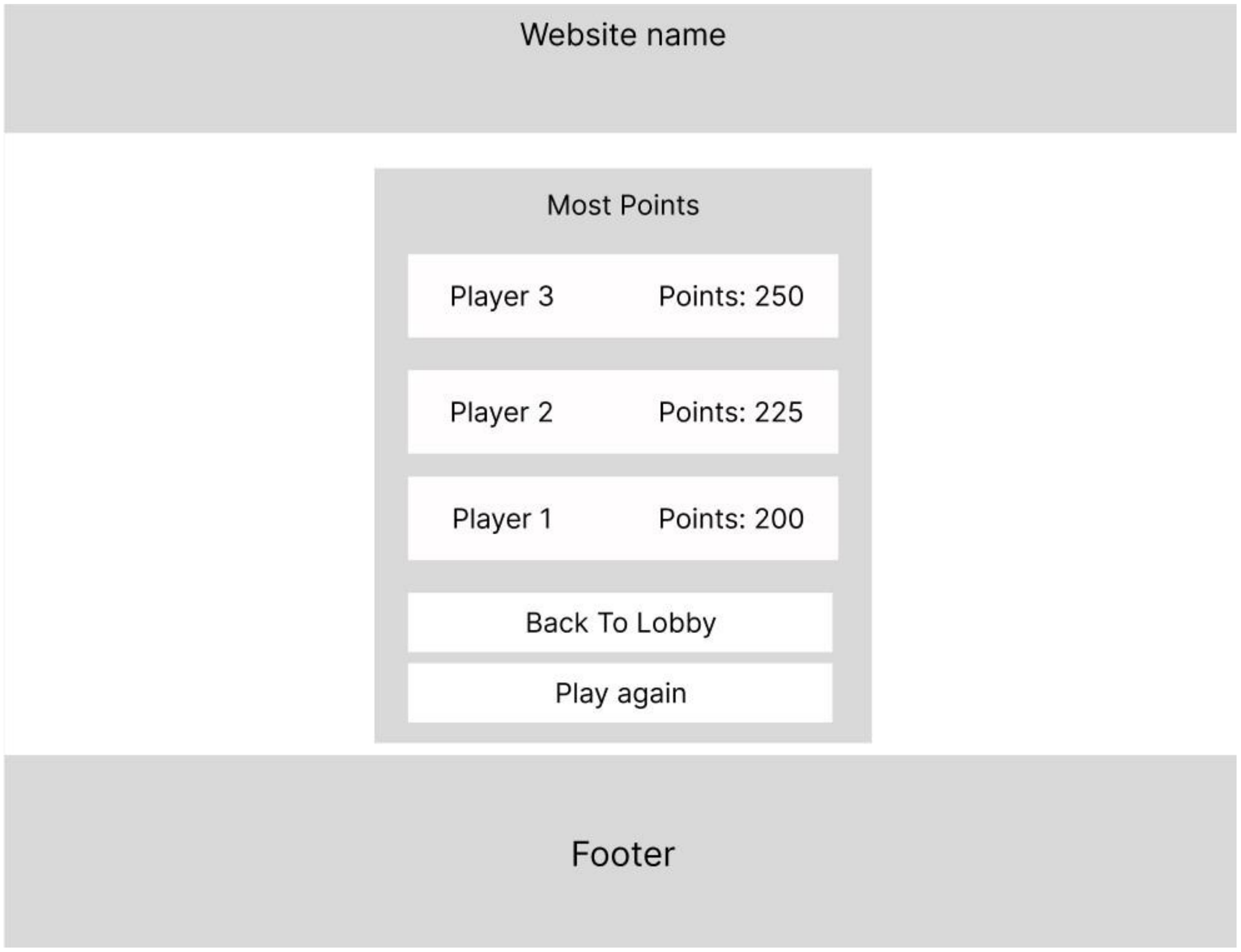


Figure 7: Score board page.

After finishing the game this wireframe is displayed.

Cloud native

Cloud native is to build and run an application using the full advantage of cloud computing. It involves developing and deploying applications in cloud environments, with a focus on scalability and automation.

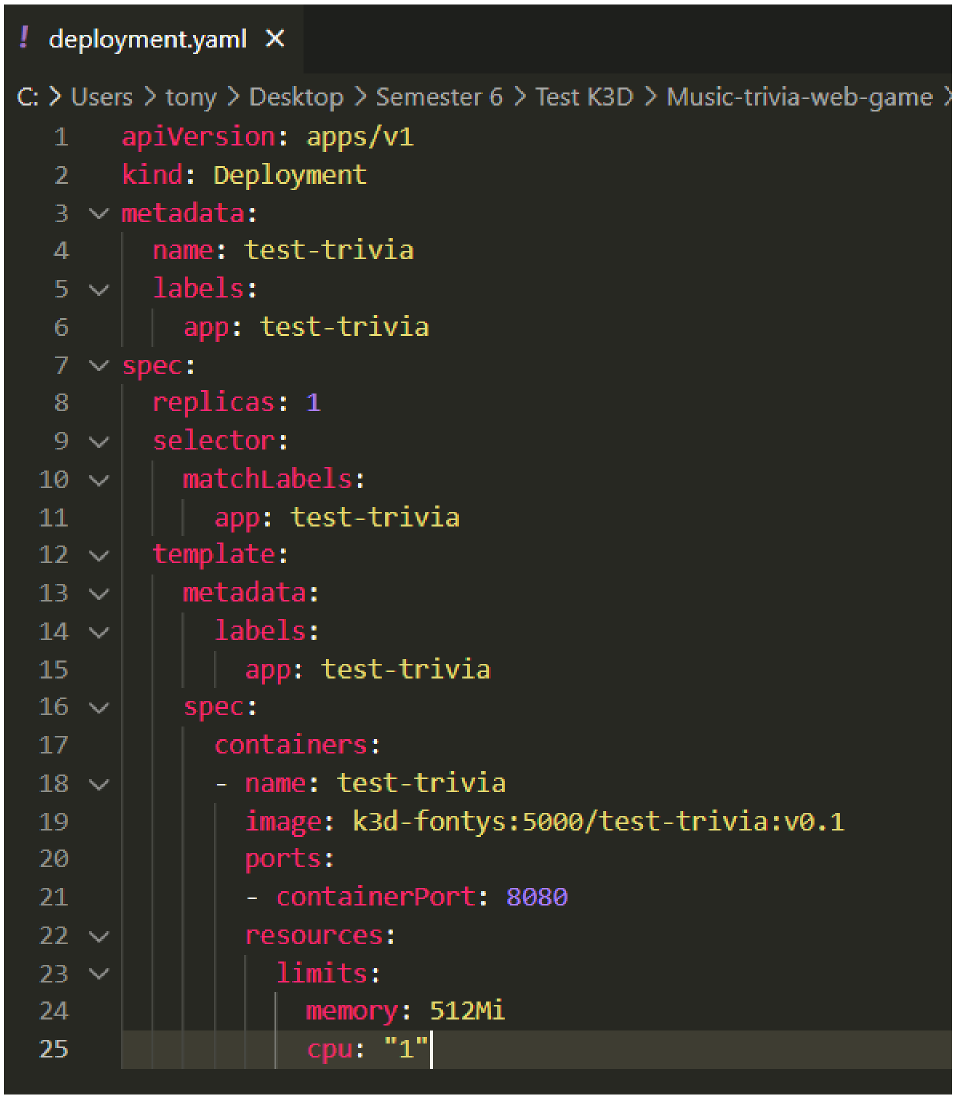
Kubernetes

Kubernetes is an open-source platform designed to automate the deployment, scaling, and operation of containerized applications. To test and learn how Kubernetes works, I'm using a tool called K3D, which is a lightweight wrapper tool for running Kubernetes clusters using Docker containerized nodes. It enables me to easily create, manage, and manipulate Kubernetes clusters using Docker.

I needed to firstly understand how Kubernetes cluster works by looking at video explanation of what Kubernetes is and then by creating my own prototype test of my backend to get an understanding of how to create a cluster and deploy it to the cluster. The deploy the backend to the cluster I need to create a node by using the “deployment.yaml” to define the node specifications.

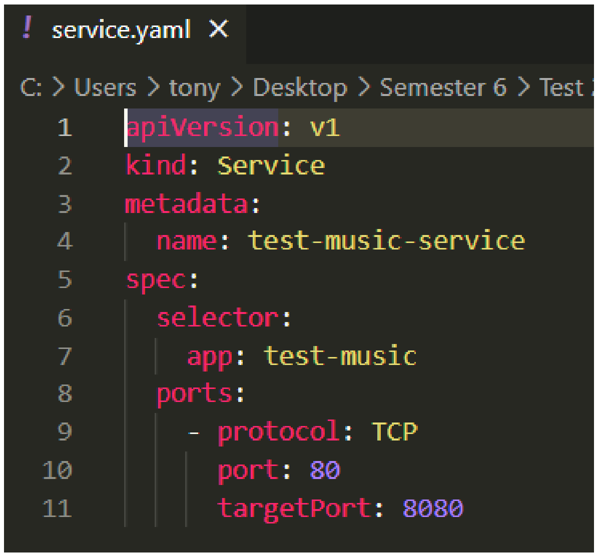
I needed to first understand how a Kubernetes cluster works by watching video explanations of what Kubernetes is and then by creating my own prototype test of my backend to gain an understanding of how to create a cluster and deploy my applications to it. To deploy the backend to the cluster, I needed to create a node by using the “deployment.yaml” file to define the node specifications.

Here is yaml specification:


deployment.yaml
C: > Users > tony > Desktop > Semester 6 > Test K3D > Music-trivia-web-game >

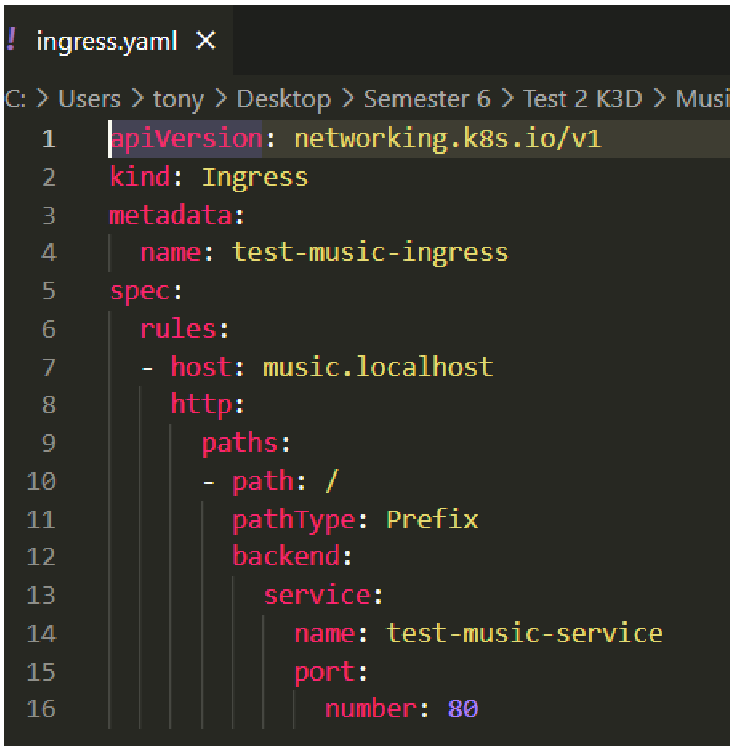
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: test-trivia
  labels:
    app: test-trivia
spec:
  replicas: 1
  selector:
    matchLabels:
      app: test-trivia
  template:
    metadata:
      labels:
        app: test-trivia
    spec:
      containers:
      - name: test-trivia
        image: k3d-fontys:5000/test-trivia:v0.1
        ports:
        - containerPort: 8080
        resources:
          limits:
            memory: 512Mi
            cpu: "1"
```

After that, create a “service.yaml” file to connect to the node. This file contains the specification of the service.



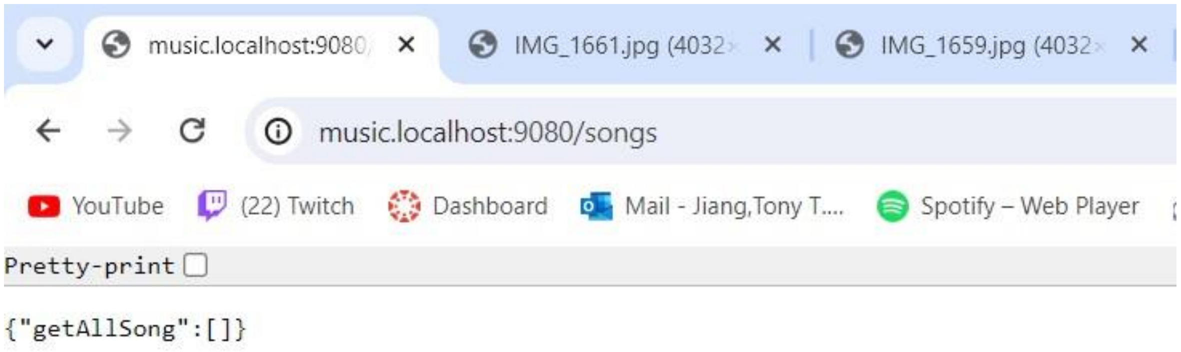
```yaml
apiVersion: v1
kind: Service
metadata:
  name: test-music-service
spec:
  selector:
    app: test-music
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
```

After that, create an “ingress.yaml” file. Ingress tells Kubernetes how traffic from external sources is routed to each service. Here are the specifications.


ingress.yaml
C: > Users > tony > Desktop > Semester 6 > Test 2 K3D > Musi

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: test-music-ingress
spec:
  rules:
  - host: music.localhost
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: test-music-service
            port:
              number: 80
```

You can connect to it using the URL and port you defined for load balancing. The URL I defined and need to connect to is http://music.localhost:9080/songs.



```
{"getAllSong":[]}
```

The next step is to connect an external database to the Kubernetes cluster. I chose this method due to time constraints. I considered adding the database to the Kubernetes cluster using a stateless approach, which seemed overwhelming and a bit difficult to implement given the time I have. If I have additional time, I wouldn't mind implementing a prototype for it.

Cloud service providers

Deploying your application to a cloud provider offers a range of infrastructure and managed services that can be leveraged for deployment and running applications. This allows you to scale your application without downtime during deployment while also managing the traffic of your application's microservices. I will write down the chosen cloud service that I'm going to use and explain why I chose it

The three most popular cloud services are:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)

Amazon Web Services (AWS)

I'm going with Amazon Web Services (AWS) because it's good cloud service and it offers 3 types free tiers depending on the service:

- Free trail that offers short free trail period.
- 12 months free.
- Always free.

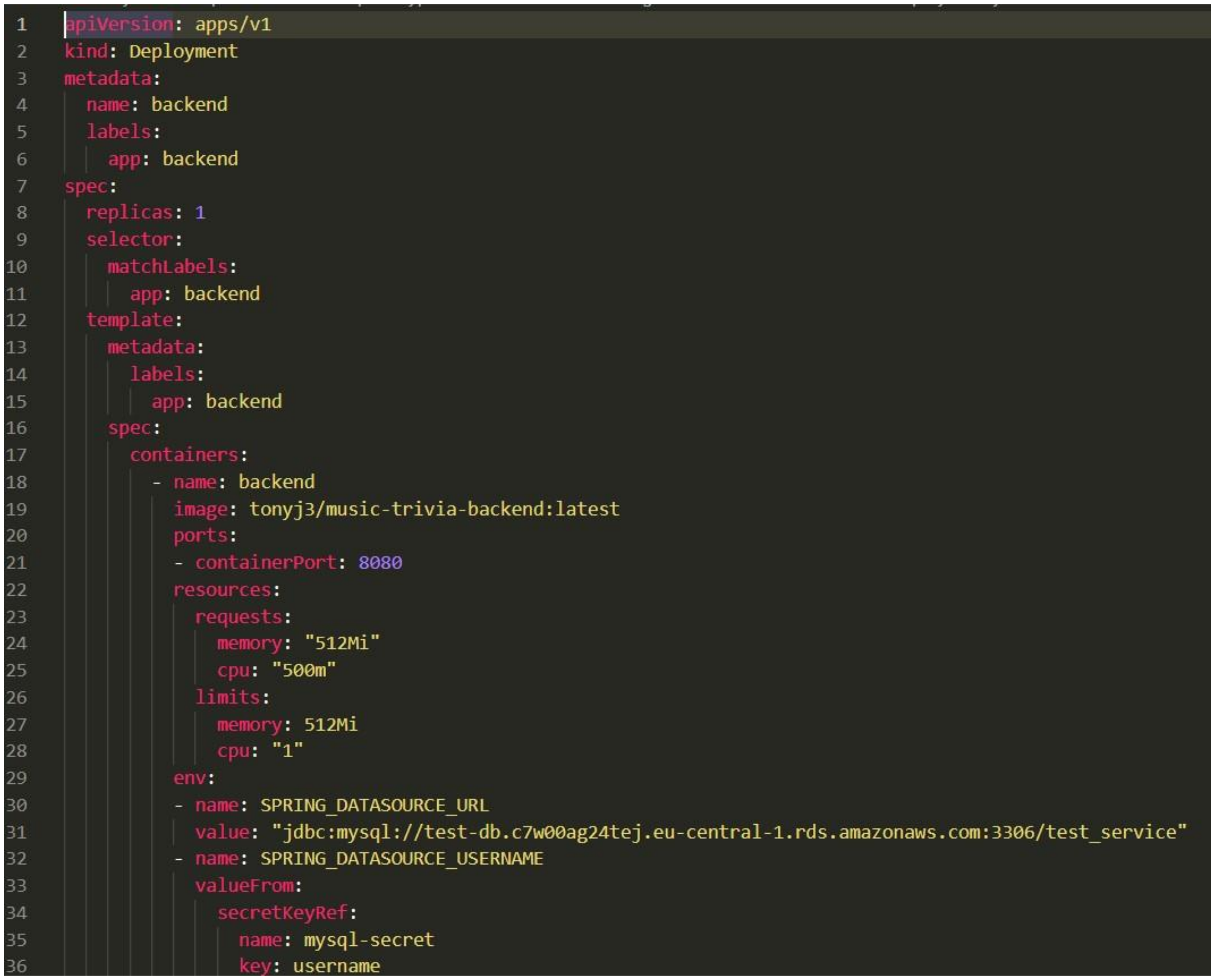
I use Amazon Elastic Kubernetes Service (EKS) to create a cluster and a tool called "eksctl" to manage it. "Eksctl" simplifies the creation, management, and deletion of EKS clusters, reducing the amount of manual configuration and setup required. I also use Amazon CloudShell to create the cluster from the command line in the cloud, so I don't have to download anything on my personal computer. Additionally, I use Ingress NGINX to manage the ingress controller, so I don't have to manage it myself.

Database connection

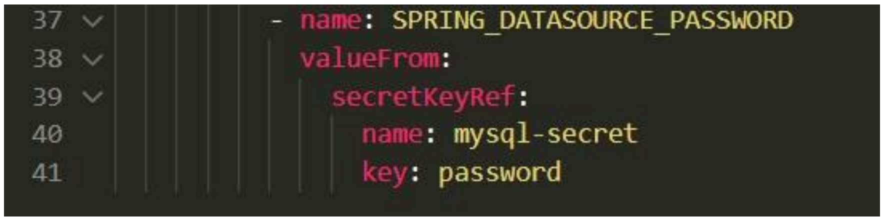
To implement the connection from Kubernetes to a MySQL database, I have decided to use AWS RDS (Relational Database Service) to connect the EKS cluster. Amazon RDS for MySQL is a managed database service that makes it easy to set up, operate, and scale a MySQL database in the cloud.

To verify if the connection to the RDS is working, I used MySQL Workbench to test it. I made the RDS instance public, configured the security group of the RDS, and adjusted the inbound and outbound rules of the associated security group to allow custom TCP port 3306. This configuration enables the connection to MySQL Workbench. However, for production environments, it's advisable to set the RDS instance to private. This enhances security, ensures compliance with regulations, and protects sensitive data from unauthorized access.

Here is how I made the connection.



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
  labels:
    app: backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
        - name: backend
          image: tonyj3/music-trivia-backend:latest
          ports:
          - containerPort: 8080
          resources:
            requests:
              memory: "512Mi"
              cpu: "500m"
            limits:
              memory: 512Mi
              cpu: "1"
          env:
          - name: SPRING_DATASOURCE_URL
            value: "jdbc:mysql://test-db.c7w00ag24tej.eu-central-1.rds.amazonaws.com:3306/test_service"
          - name: SPRING_DATASOURCE_USERNAME
            valueFrom:
              secretKeyRef:
                name: mysql-secret
                key: username
```



```yaml
- name: SPRING_DATASOURCE_PASSWORD
  valueFrom:
    secretKeyRef:
      name: mysql-secret
      key: password
```