Installation Manual for TuneIn Application

This manual provides step-by-step instructions to set up and run the TuneIn application locally. The application consists of a frontend built with React Native and a backend built with NestJS.

Prerequisites

1. **Node.js**: Ensure you have Node.js installed (version 20.0.6).
2. **npm**: npm is included with Node.js.
3. **Docker**: Install Docker for managing local databases (optional).
4. **PostgreSQL**: If not using Docker, install PostgreSQL on your machine.
5. **AWS Account**: Necessary for S3 and Cognito configurations.

Expo CLI: Install Expo CLI globally using:

```
npm install -g expo-cli
```

Clone the Repository

```
git clone https://github.com/COS301-SE-2024/TuneIn
```

```
cd TuneIn
```

Setup Backend

Step 1: Navigate to Backend Folder

```
cd backend
```

Step 2: Install Dependencies

```
npm install
```

Step 3: Configure Environment Variables

Create a `.env` file in the `backend` folder with the following format:

```
AWS_S3_BUCKET_NAME="<your-s3-bucket-name>"
```

```
AWS_S3_REGION="<your-s3-region>"


# Auth settings
JWT_SECRET_KEY="<your-jwt-secret>"
JWT_EXPIRATION_TIME="2h"
EXPO_USER_POOL_ID="<your-expo-user-pool-id>"
EXPO_CLIENT_ID="<your-expo-client-id>"
PERSONAL_EMAIL="<your-email>"
PERSONAL_PASSWORD="<your-password>"
SOCKET_ROOM_ID="<your-socket-room-id>"
SOCKET_SENDER="<your-socket-sender-id>"


# Spotify
SPOTIFY_CLIENT_ID="<your-spotify-client-id>"
SPOTIFY_CLIENT_SECRET="<your-spotify-client-secret>"
SPOTIFY_REDIRECT_URI="http://localhost:3000/auth/spotify/callback"


AUTH_STATE_SECRET_KEY="<your-auth-state-secret>"
SALT="<your-salt>"
TUNEIN_USER_ID="<your-tunein-user-id>"
```

Step 4: Setup Database

If using PostgreSQL, ensure your database is running and create the necessary tables:

Using Docker:

```
docker run --name tunein-db -e POSTGRES_USER=<your-username> -e
POSTGRES_PASSWORD=<your-password> -e POSTGRES_DB=initial_db -p
5432:5432 -d postgres
```

Using Local PostgreSQL:

- Create a database named `initial_db` and configure the user and password as specified in your `.env` file.

Step 5: Run Database Migrations

Run the following command to set up the database schema:

```
npx prisma migrate dev
```

Step 6: Start the Backend Server

Run the backend server:

```
npm run start
```

Setup Frontend

Step 1: Navigate to Frontend Folder

Open a new terminal and run:

```
cd frontend
```

Step 2: Install Dependencies

```
npm install
```

Step 3: Configure Environment Variables

Create a `.env` file in the `frontend` folder with the following format:

```
DATABASE_URL="postgresql://<your-username>:<your-password>@<your-db-host>:5432/initial_db?schema=public"


AWS_COGNITO_CLIENT_ID="<your-cognito-client-id>"

AWS_COGNITO_USER_POOL_ID="<your-cognito-user-pool-id>"
```

```
AWS_ACCESS_KEY_ID="<your-aws-access-key-id>"
AWS_SECRET_ACCESS_KEY="<your-aws-secret-access-key>"
USE_PRODUCTION_SERVER="false"
JWT_SECRET_KEY="<your-jwt-secret>"
JWT_EXPIRATION_TIME="60m"
EXPO_USER_POOL_ID="<your-expo-user-pool-id>"
EXPO_CLIENT_ID="<your-expo-client-id>"
AWS_S3_BUCKET_NAME="<your-s3-bucket-name>"
AWS_S3_REGION="<your-s3-region>"
AWS_S3_ENDPOINT="<your-s3-endpoint>"

# Spotify
SPOTIFY_CLIENT_ID="<your-spotify-client-id>"
SPOTIFY_CLIENT_SECRET="<your-spotify-client-secret>"
SPOTIFY_REDIRECT_URI="http://localhost:3000/auth/spotify/callback"
SPOTIFY_REDIRECT_TARGET="http://localhost:3000/auth/spotify/callback"
```

Step 4: Start the Frontend Application

You can run the app on a browser or an Android emulator:

- **For Browser**:

  ```
  npm start
  ```

- **For Android Emulator**:

  ```
  npm run android
  ```