

# Addis Software Test Project - MERN Stack

As a full stack developer you should be able to write both client side and server side codes. This test project will determine your ability to do that by using the MERN (MongoDB, ExpressJS, ReactJS & NodeJS) stack. Develop your frontend as well as your backend applications based on their corresponding instructions and integrate them with each other to make it a Full Stack Application.

## Backend

Develop a Rest API that will let you manage information for songs. Your API will be able to create, list, update and remove songs. You will only need to create one model to handle the song data. The expected information are

- ☐ Title
- ☐ Artist
- ☐ Album
- ☐ Genre

You are also expected to generate overall statistics:

- ☐ Total # of songs, artists, albums, genres
- ☐ # of songs in every genre
- ☐ # of songs & albums each artist has
- ☐ # songs in each album ... and so on.

You can add any stat you can think of.

## Technologies

Use the technologies below to create your backend Rest API.

- **ExpressJS** : To handle your requests.
- **MongoDB** : To store your data.
- **Mongoose** : To interact with MongoDB, model your data and to create your schema.
- **Docker**: package your backend using docker

## Frontend

The frontend part of your app should primarily show a list of songs and enable clients to create, update, and delete songs. In addition, it should have a section where the statistics data is shown. It should utilize the backend Rest API you built to fulfill the functionalities.

## Technologies

Use the technologies below to create your frontend web application.

- [Typescript](#) : use typescript to write your codes. Try not to use the type Any as much as possible.
- [ReactJS](#): To build your user interface.
- [Redux Toolkit](#): To manage the state of your app. You should first read about [redux](#) to learn the core concepts.
- [Redux - Saga](#): To make calls to your Rest API.
- [Emotion](#) and [Styled system](#): To style your app.
- Whenever a song is added, updated, or deleted, the latest changes should be shown without reloading the page.

## Bonus

Add a filtering functionality such as filter by genre. Hosting your application on free platforms like [netlify](#) or [vercel](#) for frontend and [render](#) for backend is a plus.