

# Introduction to Software Systems

## Project Phase 1: HTML/CSS Interface

Submission Due Date: 07 Apr 2023

### Concerned TAs

Sanjai Kumaran P ([sanjai.kumaran@research.iiit.ac.in](mailto:sanjai.kumaran@research.iiit.ac.in))

Vishva Saravanan R ([vishva.saravanan@research.iiit.ac.in](mailto:vishva.saravanan@research.iiit.ac.in))

### Instructions

- Maximum Marks: 100
- For queries, reach out **only** to the concerned TAs via Moodle **under the announcement thread of this project phase**.
- Submissions should be made only on the GitHub repository.
- You are allowed to make reasonable assumptions wherever you feel necessary, but you are required to document your assumptions in a file named **ASSUMPTIONS.md** in your repository.
- Since this is a team project, your repository's commit history will be checked during evaluation to make sure both members contributed equally to the work. So make sure both members commit to the repo from their own accounts.

## Overview

The objective of the project is to design a web portal for an **Internet Music Database**. This website has information about various artists, albums, and songs, with features like search, the ability to rate songs and view existing ratings, to comment on songs and view existing comments, etc.

For Phase 1, you'll have to design the base interface of the website and display information about different artists, albums, and songs of your choice using pure HTML and CSS.

## Requirements

### 1. Name your service! (5 marks)

Come up with a cool name for your music database and make sure to display it somewhere on the website.

### 2. Artists page (40 marks)

Implement an Artists page to show a list of (at least 5) of your favorite artists. **(20 marks)**  
Each artist must have the following details: **name**, **image**

Each artist item must be clickable, and clicking on an artist should redirect the user to a page listing their albums.

#### Albums page (10 marks)

Implement Albums pages, unique to each artist, to show a list of (at least 5) albums of the chosen artist.

Each album must have the following details: **name**, **image**, **number of songs**, **year of release**

Each album item must be clickable, and clicking on an album should redirect the user to a page listing its songs.

#### Songs page (10 marks)

Implement Songs pages, unique to each album, to show a list of (at least 5) songs of the chosen album.

Each song must have the following details: **name**, **duration**

The song items should not be clickable.

### 3. Home page (20 marks)

Implement a Home page with Top Charts content, i.e. with three sections: Top Artists, Top Albums, and Top Songs. **(10 marks)**

Under each section, display exactly 3 items of the corresponding section (i.e. 3 Artists under the Top Artists section, and so on).

Upon clicking an artist item from the Top Artists section, the user needs to be redirected to the page listing the albums of the chosen artist (same behavior as described in the requirements for the Artists page). **(5 marks)**

Upon clicking an album item from the Top Albums section, the user needs to be redirected to the page listing the songs in the chosen album (same behavior as described in the requirements for the Albums page). **(5 marks)**

#### 4. About page (5 marks)

Implement an About page with some details about the site () and its creators (you) ().

Examples of site details: **purpose**, **what it contains**, **what it can be used for**, etc.

Examples of creator details: **name**, **email**, **image**, etc.

Some examples of a good ‘about’ page:

- [Medium](#)

#### 5. Navigation Bar (15 marks)

Display a persistent/fixed navigation bar on the top of your page. **(10 marks)**

‘Persistent’ here refers to the fact that your navigation bar must **always** be visible on the screen regardless of how far the user has scrolled down the page.

The navigation bar should contain links to at least three pages: the Home page, the Artists page, and the About page. **(3 marks)**

Display indicators/visual cues to indicate which page the user is currently on. **(2 marks)**

Some examples of a persistent navigation bar:

- [Medium](#)
- [Stack Overflow](#)

#### 6. Footer (5 marks)

Display a small, consistent footer at the end of every page with some text of your choice. ()

‘Consistent’ here refers to the fact that all your pages must display the **same** footer contents.

In addition, implement a link/button in the footer that redirects to the ‘about’ page. ()

#### 7. Documentation (10 marks)

You’re expected to include a README.md detailing your project directory structure and instructions on how to open the website for evaluations.

Some examples of minimal footers on existing websites:

- [IIIT](#)
- [IMDb](#)

- [GitHub](#)

## Code Organization

You're expected to write clean, neatly formatted, and human-readable code. You're allowed to use tools and extensions like [Prettier](#) to help you with this.

Make sure to follow a simple and clean directory structure, for example:

```
.
|-- ASSUMPTIONS.md
|-- README.md
|-- src/
    |-- index.css
    |-- index.html
    |-- about.html
    |-- artists.html
    |-- artist01-albums.html
    |-- ...
    |-- album01-songs.html
    |-- ...
```

## Restrictions

No external packages or libraries are allowed, at all. Usage of JavaScript is not allowed for this phase either. You're expected to write everything using raw HTML and CSS.

It is up to you to intelligently design your code to minimize the amount of work you need to put in and reduce code duplication.

## Resources

- [Live Server for VS Code](#)
- [Implementing multiple linked pages in HTML](#)
- [Implementing fixed navigation bars](#)
- [Tutorial for CSS Grid Layouts](#)