

PRACTICE ASSESSMENT 7

Task: Using the starter code, build a React application that allows a user to add to a list of friends and to browse a list of songs. The application uses routing, and has three pages: a homepage with a list of friends, another page with a list of songs, and a details page that shows information about a particular song.

Setup and Submission:

See readme.md for setup instructions. Commit and push to submit. **NOTE:** The React Router npm package has already been installed. No additional installs are required.

Build Specifications:

The application consists of five components in addition to the App component. It also has two interfaces for the model.

Friend Interface

- Has two properties: **name** (string), **favoriteSong** (string)

Song Interface

- Has four properties: **id** (number), **title** (string), **artist** (string), **favorite** (boolean)
- Also export an array of hard-coded Songs from a **songs.ts** file. Use this data. (You can also copy-paste this from readme.md since copy from PDF can have issues.)

```
const songs: Song[] = [
  { id: 1, title: "Hello", artist: "Adele", favorite: false },
  { id: 2, title: "Shake It Off", artist: "Taylor Swift", favorite: true },
  { id: 3, title: "Yeah!", artist: "Usher", favorite: true },
  { id: 4, title: "Stayin' Alive", artist: "Bee Gees", favorite: false },
  { id: 5, title: "Hey Jude", artist: "Beatles", favorite: true },
  { id: 6, title: "White Christmas", artist: "Bing Crosby", favorite: false }
]
```

App Component

- Set up routing.
- At path / display the **FriendListRoute** component.
- At path /song-list display the **SongListRoute** component.
- At path /song/:id display the **SongRoute** component.
- At the top of the page, create a <header>. Include the title "**Friends & Songs App**". Also include a <nav> with the following two links. Use a <Link> tag for the links.
 - a link to the / route labeled "**Friend List**"
 - a link to the /song-list route labeled "**Song List**"

continued on next page...



FriendListRoute Component

- Displays the heading "**Friends**"
- Displays an unordered list of friends, including the name and favorite song of each.
- The data for the list comes from an array in the *state* of **FriendListRoute**.
- You may add some sample starting values to the array if you wish.
- **FriendListRoute** also displays a **FriendForm** component, which is used to add friends to the array.

FriendForm Component

- Renders a form with an input for name, an input for favorite song, and a submit button.
- Takes an **onSubmit** callback prop. **onSubmit** has one parameter, a **Friend** with **name** and **favoriteSong** values from the form.
- When the form is submitted it should call the **onSubmit** callback.
- When the form is submitted it should also clear the form, setting both inputs to empty.
- In **FriendListRoute** handle the **onSubmit** callback and add the **Friend** to the array.

SongListRoute Component

- Displays the heading "**Songs**"
- Displays an HTML table of songs, including three columns: Title, Artist, and Favorite.
- The rows of this table must be rendered using the **SongTableRow** component (see below).
- The data for the table comes from importing the array from **songs.ts** (see above).

SongTableRow Component

- Takes a **song** prop of type **Song**.
- Renders a single table row with three cells to display the title, artist, and favorite of that song.
- In the favorite cell, display a heart if **favorite** is true. If **favorite** is false, display nothing in this cell. (Hint: Use `♥` in your JSX to get a heart.)
- Use CSS to make the heart 200% font-size and PaleVioletRed in color.
- The song title should be a `<Link>` to the route (`/song/:id`) for the song.
- **SongListRoute** must use **SongTableRow** when rendering the table.

SongRoute Component

- Gets the song id from the URL param.
- Import the songs array from **songs.ts**. Use it to find the **Song** object that matches the id.
- Displays the song title, artist, and "Yes" or "No" for favorite.
- If there is no matching song, display simply: "Not Found".



Example Screenshots:

Here are sample screenshots to illustrate the task.

/ route (Friend List)

Friends & Songs App

[Friend List](#) [Song List](#)

Friends

- Rachel (favorite song: With or Without You)
- Joey (favorite song: I want a New Duck)

Name

Favorite Song

/song-list route (Song List)

Friends & Songs App

[Friend List](#) [Song List](#)

Songs

Title	Artist	Favorite
Hello	Adele	
Shake It Off	Taylor Swift	♥
Yeah!	Usher	♥
Stayin' Alive	Bee Gees	
Hey Jude	Beatles	♥
White Christmas	Bing Crosby	

/song/2 route

Friends & Songs App

[Friend List](#) [Song List](#)

Song: Shake It Off

Artist: Taylor Swift

Favorite: Yes

