

UpNext Execution Instructions

Our app is a web app that is hosted on firebase. We developed this app and tested it for using Google Chrome. To access our app, follow these steps...

1. Download Spotify Desktop here: <https://www.spotify.com/us/download/other/>
2. Go to the web address <https://upnextdebug.firebaseio.com>

If you have trouble accessing the web address or have further questions or want the database cleared, send an email to duffy18@purdue.edu

The web address provided is the buggy version of our app. We encourage you to go to <https://upnext-efec3.firebaseio.com> to see the intended behavior of the app. Keep in mind that when seeding the app with bugs, we spotted more bugs that we initially found so use the link above as a general reference to how the app should flow and not as the “solution” to finding the seeded bugs.

*Please note that this app uses a playlist name called “UpNextPlaylist”. This playlist is created upon logging in. You can use this as a way to confirm some bugs.

Below are the components and their expected behaviors:

Header:

The header is the navigation bar of our app. It is a greyish-black bar located at the top of every web page of our app.

Input	Output
User clicks “UpNext” in the header.	The app should navigate to the home page.
User clicks “Home” in the header.	The app should navigate to the home page.
User clicks “Search” in the header.	The app should navigate to the search page.
User clicks “Login” in the header.	The app should navigate to the login page.
Mobile version, user clicks the button to expand the menu.	The menu should expand. On mobile, the menu should close when a tab is chosen.

Home:

The home page is the heart of our app. It is the first page you should encounter. Here users if logged in can listen to the shared playlist on repeat and also vote on the playlist's songs.

Input	Output
User clicks "Please Login" button.	The app should navigate to the login page.
User clicks the play button in the embedded iframe.	The Spotify Desktop app should open, music should start playing, and a banner displaying what song is currently playing should pop up.
User clicks one of the voting buttons.	The number to the right of a song's title should adjust accordingly. The order of the songs is based on the amount of votes.
User clicks the pause button in the embedded iframe.	The song should pause.
User clicks the skip forward button in the embedded iframe.	The next song should start playing.
User clicks the skip back button in the embedded iframe.	The previous song should start playing. This feature only applies to premium spotify users. All others should be notified why they cannot go back a song.

Login:

The login page is a bare bones means of initiating OAuth2 login through spotify.

*Please note that the absence of a logout button is NOT a bug. We never got to implementing this.

Input	Output
User clicks the "Login Here" button	A new window should pop up prompting the user to enter their username and password. (Keep in mind that an auth token is generated. We store it so login is faster after the first time. This is done to skip past entering credentials)
User successfully logs in	The spotify pop up window should close and the app should navigate to the home page

	with all elements loaded including the iframe.
--	--

Search:

The search page is where users can search the spotify track library. They can then add songs to the UpNextPlaylist to listen to on the web app.

Input	Output
User clicks the "Search" button	A list of terms should show up with a green plus button. (Very odd behavior with inputs may be considered a bug). Please note that the list returned is entirely determined by the Spotify API.
User clicks the green plus button by a song name.	The song should be added to the playlist as long as there is not a track with an identical track id.

Other:

If you find a bug other than what is covered by the behavior listed above send an email to duffy18@purdue.edu to check with me if this was one of the seeded bugs.