Take Home Exercise 3

Evangeline Olivia, Patricia Trisno, Xu Xinyi

# Prototype Module for Shiny Application

## The Task

In this take-home exercise, we are required to select one of the modules of our proposed Shiny application (Group Project) and complete the following tasks

* To evaluate and determine the necessary R packages needed for our Shiny application are supported in R CRAN,
* To prepare and test that the specific R codes can run and returns the correct output as expected,
* To determine the parameters and outputs that will be exposed on the Shiny applications,
* To select the appropriate Shiny UI components for exposing the parameters determined above, and
* To include a section called UI design for the different components of the UIs for the proposed design.

## Getting Started

Our project will be using data from the [VAST Challenge 2025](https://vast-challenge.github.io/2025/index.html) Mini-Challenge 1.

Specifically, our project will focus on Mini-Challenge 1, where participants are provided with a knowledge graph containing recording artists, albums, collaborations, concerts, influences, and more. The goal is to develop visual analytics approaches to trace the rise of one artist and predict who will be the next to break out.

In addition, we plan to develop the following Shiny applications:

* **Shiny App 1: Sailor Shift’s Career Explorer**
* **Shiny App 2: Oceanus Folk Influence Tracker**
* **Shiny App 3: Rising Star Prediction Dashboard**

For this take-home exercise, we will be providing the prototype module report and the UIs designed for the above applications.

## R Packages

pacman::p\_load(tidyverse, jsonlite,  
 tidygraph, ggraph, igraph, plotly, visNetwork,ggtext, ggiraph,patchwork,  
 kableExtra, showtext,lubridate)

## Dataset

data <- fromJSON("data/MC1\_graph.json")

## Storyboard

The storyboard is designed to visually represent the user’s journey within the R Shiny dashboard applications. It acts as a bridge between research insights and user interaction, illustrating how findings are translated into visual and interactive elements. The UI design and interactive features are thoughtfully developed to enable seamless exploration and analysis of data including network components helping users identify key patterns and trace the rising of artist in the music industry.

The prototype is organized into three main focus areas:

* Sailor Shift’s Career Explorer
* Oceanus Folk Influence Tracker
* Rising Star Prediction Dashboard

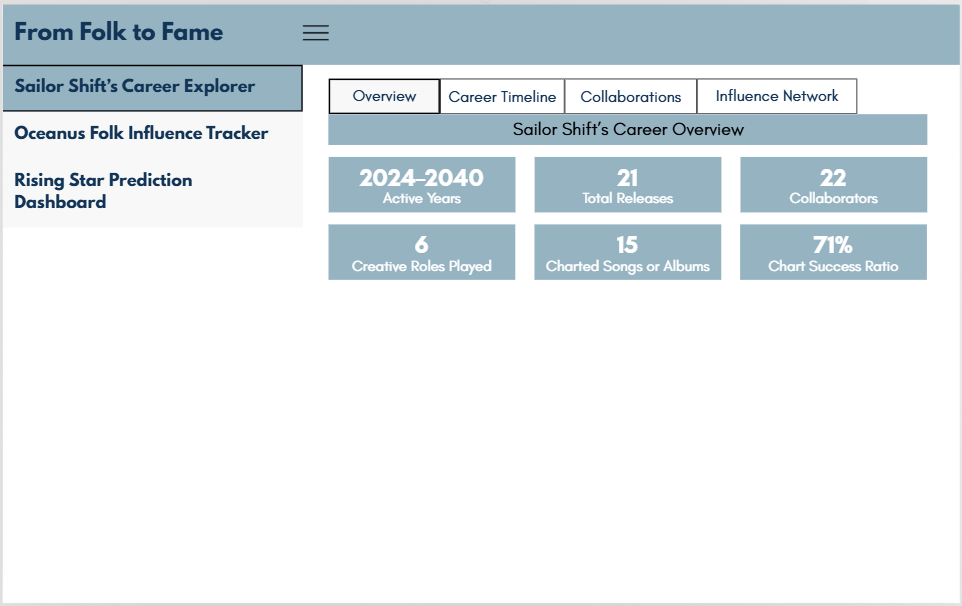
## Sailor Shift’s Career Explorer

### Section One - Overview

This tab serves as the landing page for Sailor Shift’s Career Explorer module. It provides a high-level summary of the artist’s musical trajectory. As shown in the figure below, the Overview sub-tab consolidates key statistics into a series of information cards that allow users to quickly grasp the scope and success of her career.

Career metrics display, showing:

* Years active
* Number of releases
* Number of collaborators
* Number of creative roles
* Number of charted works
* Chart success ratio



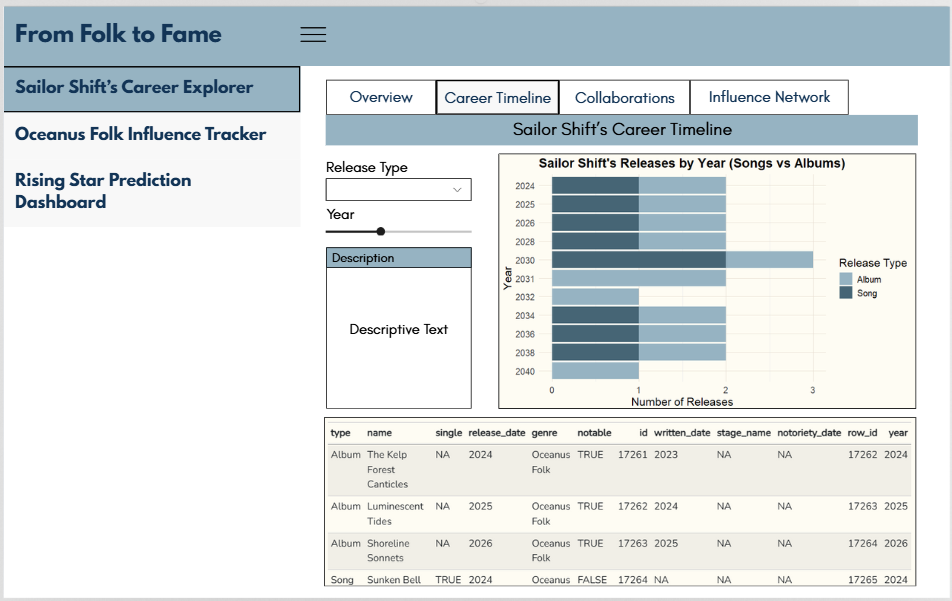
Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI for Overview tab - Sailor Shift’s Career Explorer  
SailorOverviewRow1 <- fluidRow(  
 tabsetPanel(  
 tabPanel("Overview",   
 fluidRow(  
 valueBox(),  
 valueBox(),  
 valueBox()  
 ),  
 fluidRow(  
 valueBox(),  
 valueBox(),  
 valueBox()  
 )  
 )  
 )  
)  
  
# Server Components  
   
output$activeYears <- renderValueBox({})  
output$totalReleases <- renderValueBox({})  
output$collaborators <- renderValueBox({})  
output$creativeRoles <- renderValueBox({})  
output$chartedWorks <- renderValueBox({})  
output$chartSuccess <- renderValueBox({})

### Section Two - Career Timeline

The Career Timeline tab enables users to interactively explore Sailor Shift’s music releases over the years. Users can:

* Filter releases by type (Album or Song)
* Select a year range using a slider
* Read description text that updates dynamically
* View a bar chart displaying the number of releases per year by type
* Examine a detailed data table of Sailor’s works with attributes like title, release year, genre, and whether the work was notable



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI   
CareerTimelineUI <- fluidPage(  
 titlePanel("Sailor Shift’s Career Timeline"),  
   
 tabsetPanel(  
 tabPanel("Career Timeline",  
 fluidRow(  
 column(  
 selectInput()),  
 sliderInput(),  
 textAreaInput()  
 ),  
 column(  
 plotOutput()  
 )  
 ),  
 br(),  
 DT::dataTableOutput()  
 )  
 )  
)  
  
#Server  
output$release\_barplot <- renderPlot({})  
output$release\_table <- DT::renderDataTable({})

### Section Three - Collaborations

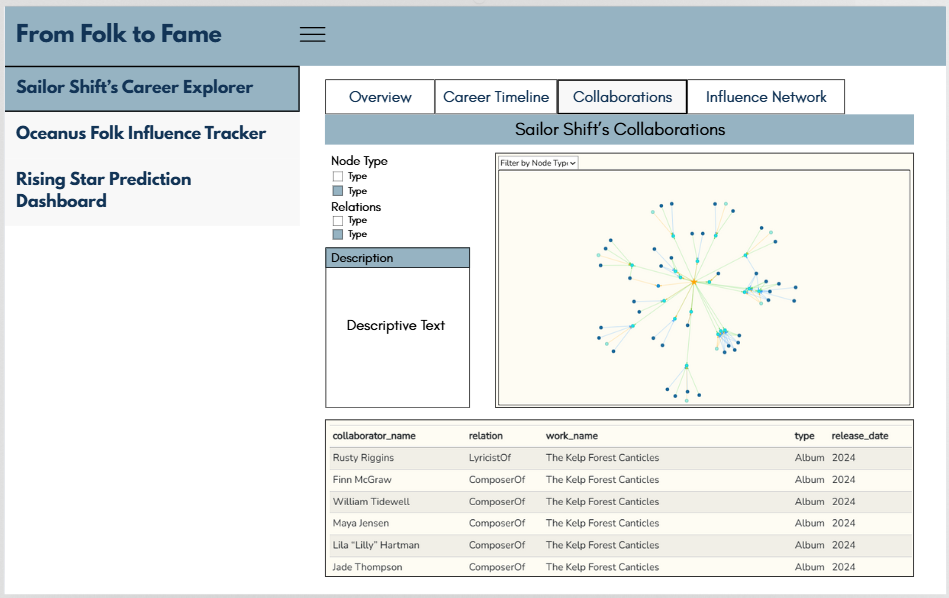
The Collaborations tab provides a detailed network-based view of Sailor Shift’s creative relationships. It includes:

Node and Relation Type Filters: These checkboxes allow users to filter the network by the type of entity (e.g., person, song, album) or relationship (e.g., ComposerOf, LyricistOf).

Interactive Network Plot: A central network visualization shows Sailor at the center, connected to collaborators through edges labeled with their creative roles.

Descriptive Panel: Offers a placeholder for dynamic text or analysis summaries.

Data Table: Lists each collaborator with their name, relation to the work, work title, type (Album/Song), and release date.



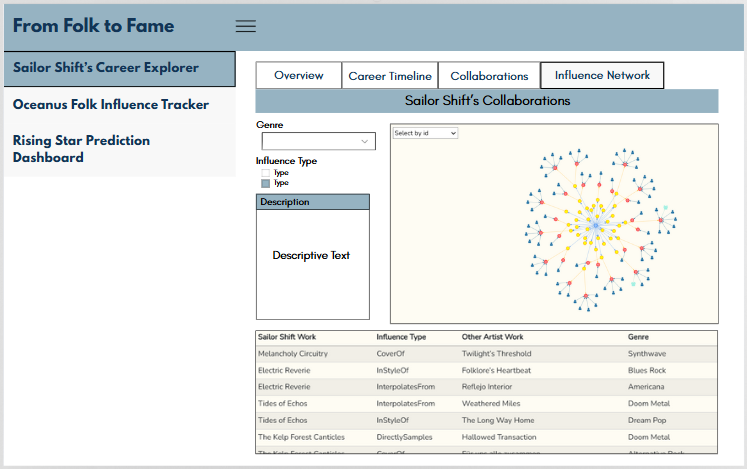
Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
SailorCollabUI <- fluidPage(  
 titlePanel("Sailor Shift’s Collaborations"),  
   
 tabsetPanel(  
 tabPanel("Collaborations",  
 fluidRow(  
 column(   
 checkboxGroupInput(), selected =),  
 checkboxGroupInput(),  
 textAreaInput()  
 ),  
 column(  
 visNetworkOutput()  
 )  
 ),  
 br(),  
 DT::dataTableOutput()  
 )  
 )  
)  
  
#Server  
output$collab\_network <- renderVisNetwork({})  
output$collab\_table <- DT::renderDataTable({})

### Section Three - Influence Network

The Influence Network tab visualizes how Sailor Shift’s musical works relate to others through influence connections. It allows users to:

* Filter by Genre (e.g., Doom Metal, Blues Rock)
* Filter by Influence Type (e.g., CoverOf, InStyleOf, DirectlySamples)
* Search and Select a Work from a dropdown list for focused exploration
* View a directed network plot that visually maps how her songs and albums connect to or are influenced by other artists’ works
* Read descriptive context in a dedicated panel
* Examine a data table listing all pairwise influence relationships, their types, and associated genres



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
InfluenceNetworkUI <- fluidPage(  
 titlePanel("Sailor Shift’s Influence Network"),  
   
 tabsetPanel(  
 tabPanel(   
 fluidRow(  
 column(   
 selectInput(, choices = c(), selected =),  
 checkboxGroupInput(  
 choices = c(),  
 selected = c()),  
 selectizeInput(),  
 textAreaInput()  
 ),  
 column(  
 visNetworkOutput()  
 )  
 ),  
 br(),  
 DT::dataTableOutput()  
 )  
 )  
)  
  
#Server  
output$influence\_graph<- renderVisNetwork({})  
output$influence\_table <- DT::renderDataTable({})

## Oceanus Folk Influence Tracker

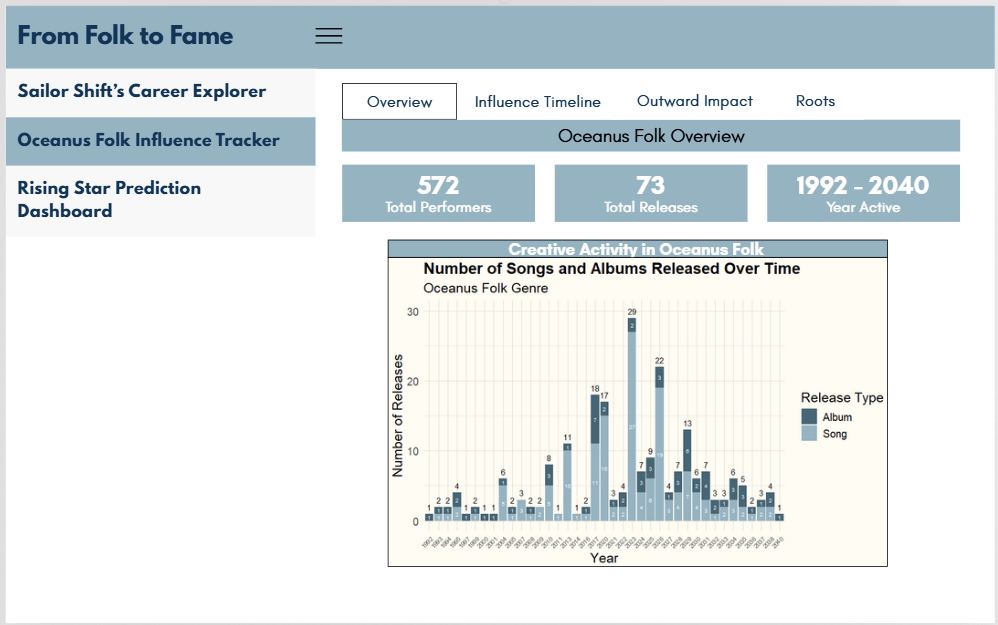
### Section One - Overview

The Overview tab within the Oceanus Folk Influence Tracker module provides users with a macro-level snapshot of the genre’s scope and evolution.

Key features include:

Metric summary cards for:

* Total performers in the Oceanus Folk community
* Total music releases
* Span of activity years
* A stacked bar chart visualizing the number of songs vs albums released over time, helping users observe patterns in creative output (e.g., genre peaks, format preferences)



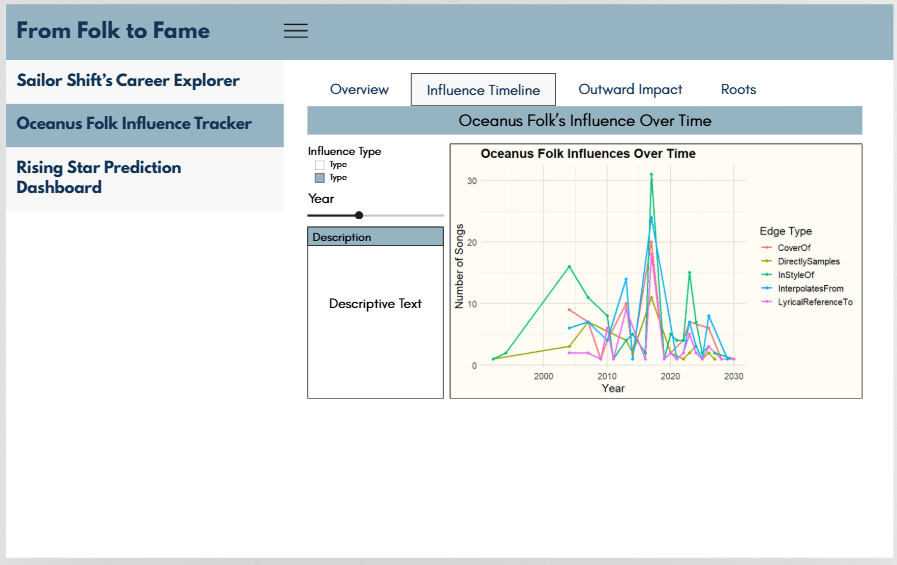
Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
OceanusFolkOverviewUI <- fluidPage(  
 titlePanel("Oceanus Folk Overview"),  
   
 tabsetPanel(  
 tabPanel(  
 fluidRow(  
 valueBox(),  
 valueBox(),  
 valueBox()  
 ),  
 br(),  
 fluidRow(  
 column(  
 plotOutput()  
 )  
 )  
 )  
 )  
)  
  
# Server  
output$totalperformers <- renderValueBox({})  
output$totalReleases <- renderValueBox({})  
output$activity\_year <- renderValueBox({})  
  
output$folk\_release\_chart <- renderPlot({})

### Section Two - Influence Timeline

The Influence Timeline tab visualizes how Oceanus Folk has contributed to musical influence across time. Key components include:

* Checkbox filters to select different influence types (e.g., CoverOf, InterpolatesFrom, LyricalReferenceTo)
* A year range slider to narrow the temporal window of interest
* A multicolored line chart showing trends in the number of influence connections (by type) over time
* A description box for narrative or insight generation



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

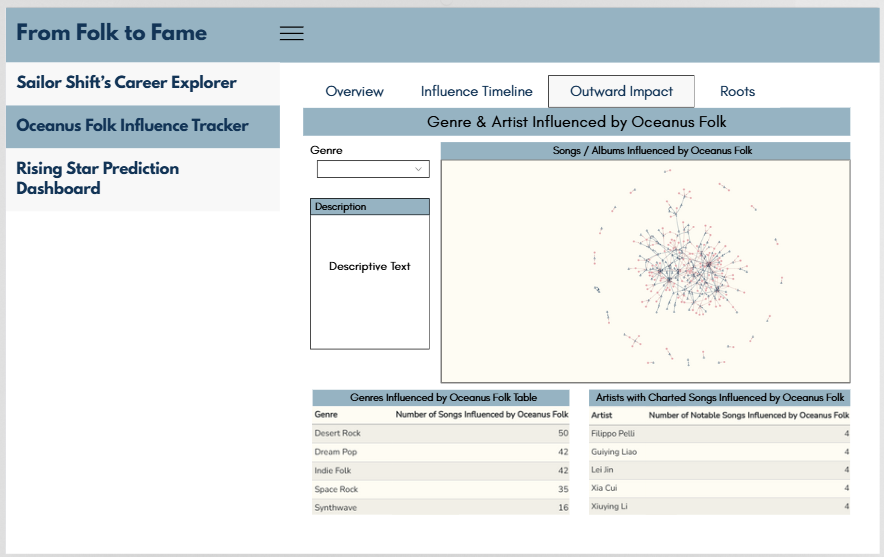
# UI  
InfluenceTimelineUI <- fluidPage(  
 titlePanel(),  
   
 tabsetPanel(  
 tabPanel(  
 fluidRow(  
 column(  
 checkboxGroupInput(   
 choices = c(),  
 selected = c()),  
 sliderInput(),  
 textAreaInput()  
 ),  
 column(  
 plotOutput()  
 )  
 )  
 )  
 )  
)  
  
# Server  
output$influence\_timeline\_chart <- renderPlot({})

### Section Three - Outward Impact

The Outward Impact tab explores the ripple effect of the Oceanus Folk genre across genres and artists. It features:

* A genre filter dropdown to narrow results by specific genres influenced
* A central network graph visualizing songs/albums influenced by Oceanus Folk across different genres and artists
* A description box for contextual notes or automated narrative summaries

Two tables: - One summarizing the number of songs per genre influenced by Oceanus Folk - Another listing artists with the most notable songs influenced by the genre



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

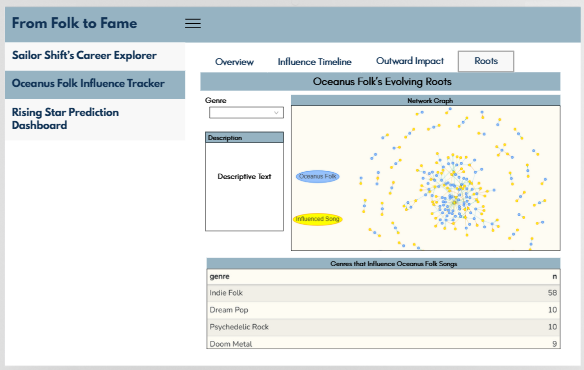
# UI  
OutwardImpactUI <- fluidPage(  
 titlePanel("Genre & Artist Influenced by Oceanus Folk"),  
   
 tabsetPanel(  
 tabPanel(   
 fluidRow(  
 column(  
 selectInput(),  
 textAreaInput()  
 ),  
 column(  
 visNetworkOutput()  
 )  
 ),  
 br(),  
 fluidRow(  
 column(  
 h5(),  
 DT::dataTableOutput()  
 ),  
 column(  
 h5(),  
 DT::dataTableOutput()  
 )  
 )  
 )  
 )  
)  
  
# Server  
output$influenced\_network <- renderVisNetwork({})  
output$influenced\_genre\_table <- DT::renderDataTable({})  
output$influenced\_artist\_table <- DT::renderDataTable({})

### Section Four - Roots

The **Roots** tab helps users uncover the foundation influences behind Oceanus Folk songs.

Users can:

* Filter the network by genre to explore specific stylistic influences
* View a network graph highlighting genres and songs that influenced Oceanus Folk, with Oceanus Folk nodes and influenced song nodes distinctly colored
* Read a dynamic description that explains the context and insights
* Analyze a table summarizing which genres most frequently influenced Oceanus Folk tracks



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
RootsTab <- fluidPage(  
 titlePanel("Oceanus Folk’s Evolving Roots"),  
   
 fluidRow(  
 column(  
 selectInput( inputId, label, choices = c()), # Dropdown to filter genre  
   
 tabsetPanel(  
 tabPanel(  
 title = "Description",  
 verbatimTextOutput("description\_text") # Descriptive explanation  
 )  
 )  
 ),  
   
 column(  
 plotOutput() # Display network visualization of genre-song influence  
 )  
 ),  
   
 fluidRow(  
 column(  
 DT::dataTableOutput() # Data table showing genre and count  
 )  
 )  
)  
  
  
# Server  
output$roots\_description\_text <- renderText({})  
output$roots\_network <- renderPlot({})  
output$roots\_genre\_table <- DT::renderDataTable({})

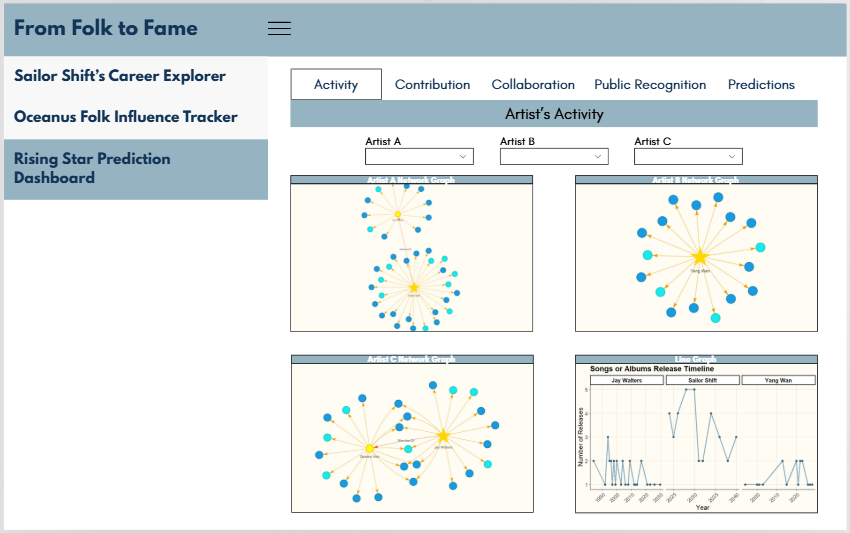
## Rising Star Prediction Dashboard

### Section One - Rising Star Prediction Dashboard

The **Activity** tab provides a comparative view of artists’ release behaviors and performance history.

Users can:

* Select up to three artists (A, B, and C) for side-by-side comparison
* View individual network graphs showing each artist’s release and influence structure
* Analyze a timeline line chart of songs or albums released by the selected artists over time
* Track activity trends to identify peak creative periods and gaps across artist careers



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

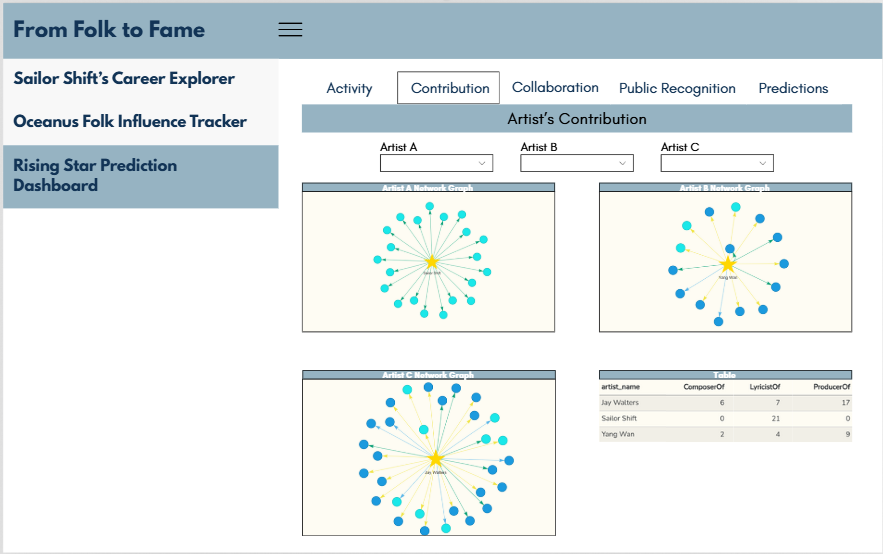
# UI  
ArtistsActivityTab <- fluidPage(  
 titlePanel("Artist’s Activity"),  
   
 fluidRow(  
 column(selectInput(choices = c()) # Dropdown for Artist A  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist B  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist C  
 )  
 ),  
   
 fluidRow(  
 column(  
 plotOutput() # Plot for Artist A  
 ),  
 column(  
 plotOutput() # Plot for Artist B  
 )  
 ),  
   
 fluidRow(  
 column(  
 plotOutput() # Plot for Artist C  
 ),  
 column(  
 plotOutput() # Line graph comparing release trends  
 )  
 )  
)  
  
# Server  
  
output$artist\_a\_network <- renderPlot({})  
output$artist\_b\_network <- renderPlot({})  
output$artist\_c\_network <- renderPlot({})  
output$release\_timeline <- renderPlot({})

### Section Two - Contribution

The **Contribution** tab highlights the creative roles played by artists in Oceanus Folk.

Users can:

* Select and compare up to three artists (A, B, and C)
* Explore network graphs that visualize each artist’s contributions as a composer, lyricist, or producer
* Analyze a contribution table showing the number of works attributed to each role
* Identify key contributors to the genre’s development through their behind-the-scenes impact



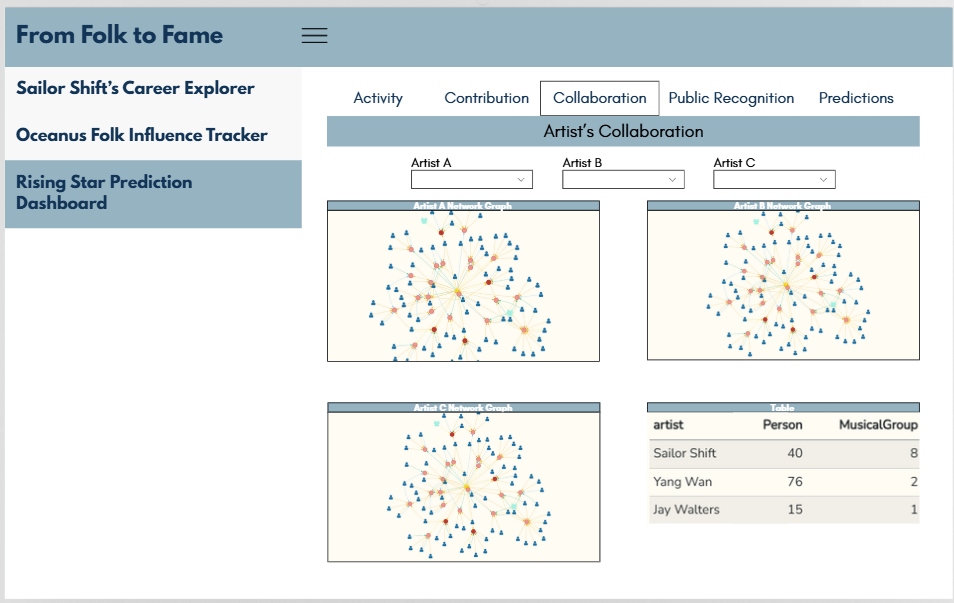
Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
ArtistsContributionTab <- fluidPage(  
 titlePanel("Artist’s Contribution"),  
   
 fluidRow(  
 column(selectInput(choices = c()) # Dropdown for Artist A  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist B  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist C  
 )  
 ),  
   
 fluidRow(  
 column(plotOutput() # Artist A Network Graph  
 ),  
 column(plotOutput() # Artist B Network Graph  
 )  
 ),  
   
 fluidRow(  
 column(plotOutput() # Artist C Network Graph  
 ),  
 column(DT::dataTableOutput() # Contribution\_table  
 )  
 )  
)  
  
# Server  
output$contrib\_graph\_a <- renderPlot({}) # Network graph for Artist A's contributions (ComposerOf, LyricistOf, ProducerOf)  
output$contrib\_graph\_b <- renderPlot({}) # Network graph for Artist B's contributions  
output$contrib\_graph\_c <- renderPlot({}) # Network graph for Artist C's contributions  
  
output$contribution\_table <- DT::renderDataTable({}) # DataTable showing ComposerOf, LyricistOf, and ProducerOf for selected artists

### Section Three - Collaboration

The **Artist’s Collaboration** tab showcases the collaborative relationships of Oceanus Folk artists. Users can:

* Select and compare up to three artists (A, B, and C)
* Visualize each artist’s collaboration network with individuals and musical groups
* Analyze differences in collaboration patterns using structured network graphs
* Refer to a summary table that lists the number of collaborations with persons and musical groups per artist



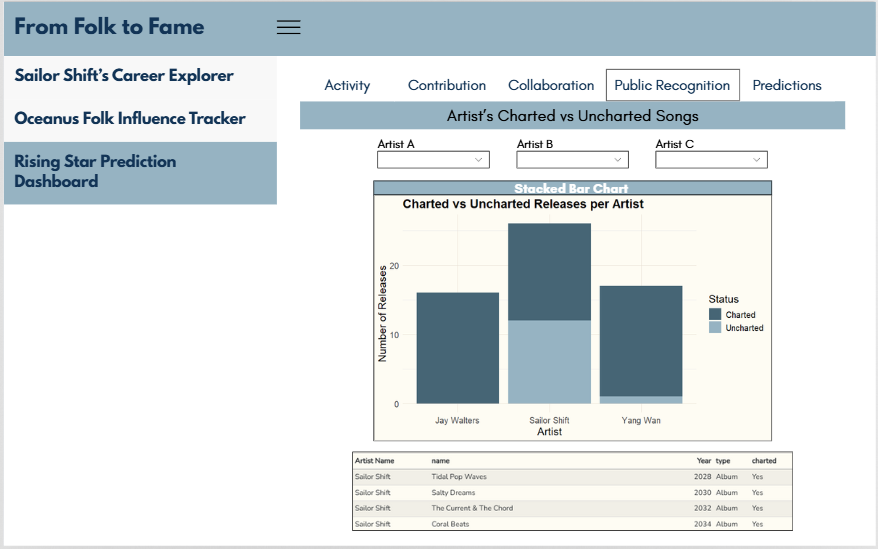
Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
ArtistsCollaborationTab <- fluidPage(  
 titlePanel("Artist’s Collaboration"),  
   
 fluidRow(  
 column(selectInput(choices = c()) # Dropdown for Artist A  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist B  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist C  
 )  
 ),  
   
 fluidRow(  
 column(plotOutput() # Artist A Network Graph  
 ),  
 column(plotOutput() # Artist B Network Graph  
 )  
 ),  
   
 fluidRow(  
 column(plotOutput() # Artist C Network Graph  
 ),  
 column(DT::dataTableOutput() # Collaboration\_table  
 )  
 )  
)  
  
# Server  
output$collab\_graph\_a <- renderPlot({}) # Network plot showing Artist A's collaborations  
output$collab\_graph\_b <- renderPlot({}) # Network plot showing Artist B's collaborations  
output$collab\_graph\_c <- renderPlot({}) # Network plot showing Artist C's collaborations  
output$collaboration\_table <- DT::renderDataTable({}) # Table showing artist + number of collaborations with Person & MusicalGroup

### Section Four - Public Recognition

The **Public Recognition** tab compares how often each artist’s work charted in public rankings. Users can:

* Select up to three artists (A, B, and C) for comparison
* View a stacked bar chart that contrasts the number of *charted* versus *uncharted* releases per artist
* Explore a detailed table listing each artist’s songs or albums along with their release year, type, and chart status
* Analyze recognition trends and identify which artists consistently achieve public acclaim



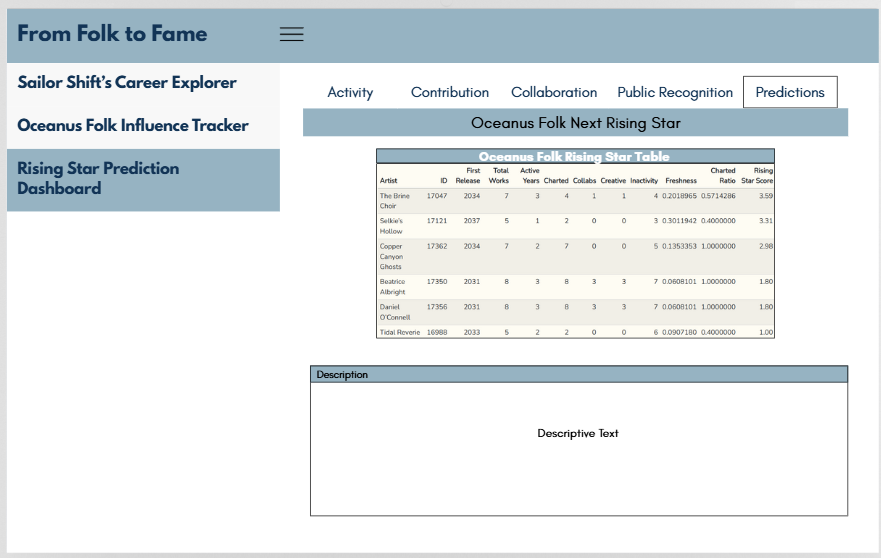
Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
  
PublicRecognitionTab <- fluidPage(  
 titlePanel("Artist’s Charted vs Uncharted Songs"),  
   
 fluidRow(  
 column(selectInput(choices = c()) # Dropdown for Artist A  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist B  
 ),  
 column(selectInput(choices = c()) # Dropdown for Artist C  
 )  
 ),  
   
 fluidRow(  
 column(plotOutput() # Stacked bar chart for releases  
 )  
 ),  
   
 fluidRow(  
 column(DT::dataTableOutput() # Detailed list of songs/albums and chart status  
 )  
 )  
)  
  
# Server  
output$charted\_vs\_uncharted <- renderPlot({}) # Create a stacked bar chart showing Charted vs Uncharted songs per artist  
output$charted\_table <- DT::renderDataTable({}) # Table with artist name, song/album, year, type, and charted status

### Section Five - Predictions

The **Predictions** tab identifies emerging talents most likely to become the next big names in Oceanus Folk. Users can:

* Explore a ranked table of rising stars, sorted by calculated “Rising Star Score”
* Review artist attributes including first release year, total works, collaborations, freshness, and inactivity
* Understand how metrics like charted ratio and creative activity contribute to the prediction
* Read a description that explains how the prediction model works and what each variable means



Code chunk below shows the simplified version of UI and Server components in R Shiny application for Overview sub-tab.

# UI  
  
PredictionsTab <- fluidPage(  
 titlePanel("Oceanus Folk Next Rising Star"),  
   
 fluidRow(  
 column(DT::dataTableOutput() # Main prediction result table  
 )  
 ),  
   
 fluidRow(  
 column(tabsetPanel(  
 tabPanel(  
 title = "Description",  
 wellPanel(verbatimTextOutput("rising\_star\_description") # Descriptive Explanation  
 )  
 )  
 )  
 )  
 )  
)  
  
# Server  
output$rising\_star\_table <- DT::renderDataTable({}) # Table with columns like: Artist, ID, First Release, Total Works, etc.  
output$rising\_star\_description <- renderText({}) # Explanation of the Rising Star Score and what users should interpret