Name: Ethan Yuen (Code Reviewer)
Group: GCIEL
Date: 4/19/2024
Document Name: app.R

App Code Review

I reviewed the code for the Navbar. Altered code is provided below.

Code Construction	Comments
Functions, Methods, and Size	Since the navbar/UI interface is a singular function utilizing CSS, this is the simplest it will get; hence, we are unable to simplify the code any further.
Formatting, Layout and Style	We utilize consistent indenting to ensure that all code looks the same; we avoid utilizing the tab key, as R automatically indents based on the function.
White Space	We use empty lines to separate code blocks, and as above we ensure that they are all consistently indented.
Block and Statement Style Guidelines	None.
Declaration Style Guidelines	None.
Commenting Style Guidelines	All comments begin with a '#' followed by a space. We did have to change several comments from '##' to '#'.
Identifier Naming Conventions	None.
Defensive Programming: Questions	N/A. Data file upload ensures that it is correct.
Error Handling	N/A

```
# Adding visualization descriptions (These are brief and need to be modernized).
# Update made 4.11.24
dynamicDescription <- function(vis){
  switch(
    vis,
    "Data" = "This table displays the uploaded dataset.",
    "Total Completion Time per Piece" = "This plot shows the total completion time per piece.",
    "Video Engagement Analysis" = "This plot analyzes video engagement by piece.",
    "3D Distance Analysis by Piece" = "This 3D plot analyzes distance by piece.",
    "3D Distance Analysis by Player" = "This 3D plot analyzes distance by player.",
    "Player Positions Heatmap" = "This heatmap visualizes player positions.",
```

```
    "Player Location Animation Draft 1" = "This animation shows player location over time.",
    "Player Location Animation Draft 2" = "This animation shows player location over time.",
    "ARCS Model Based Evaluation" = "This section provides an ARCS model-based evaluation."
  )
}


# Defining the user interface
ui <- shinyUI(
  fluidPage(
    shinyjs::useShinyjs(),
    shinyjs::inlineCSS(css),

    # Our custom HTML/CSS for the UI theme
    tags$head(
      tags$style(
        HTML(
          '
        }
        .animated-title {
          animation: fadeIn 1s; /* Use fadeIn animation for 1 second */
        }
        @keyframes fadeIn {
          from {
            opacity: 0;
          }
          to {
            opacity: 1;
          }
        }
      '
        )
      )),

    # Navbar Page
    navbarPage(
      title = "GCIEL Assessment Strategy",
      tabPanel(
        "Input Data",

        # Instructional text for users
        HTML(
          '<p style="color: #aa66cc;">Please ensure to review the data
          description before uploading a CSV file. Make sure that the dataset
```

```
    you upload matches the data description in order for the app to work
    correctly.</p>'
  ),

  fileInput(
    "file",
    label = "Upload your dataset (CSV)",
    accept = c("text/csv"),
    multiple = FALSE,
    width = "80%"
  ),

  # Data PDF Download Link
  downloadLink(
    "dataDescriptionLink",
    "Download Data Description",
    class = "download-link"
  ),

  br(),

  # Data CSV Download Link
  downloadLink(
    "vikingshipDataLink",
    "Download Data (CSV)",
    class = "download-link"
  ),

  # Data Table Output
  renderDT(
    "outFile",
    options = list(
      lengthChange = TRUE,
      pageLength = 10
    )),

  dataTableOutput("outFile")
),

tabPanel(
  "Data Vis",

  # Specify layout for side panel.
  # Side panel is used for modifying graphs using drop down bars and sliders.
```

```
sidebarLayout(
  sidebarPanel(
    # Side panel for time graph
    conditionalPanel(
      # Check the  graph is selected
      condition = "input.tabs == 'use' && input.tabs_use == 'completion'",
      ns = NS(NULL),

      # Panel title
      h3("Total Completion Time per Piece"),
      # Context
      h5(dynamicDescription("Total Completion Time per Piece"))
    ),

    # Side panel for time graph
    conditionalPanel(
      # Check the  graph is selected
      condition = "input.tabs == 'use' && input.tabs_use == 'video'",
      ns = NS(NULL),

      # Panel title
      h3("Video Engagement Analysis"),
      # Context
      h5(dynamicDescription("Video Engagement Analysis"))
    ),

    # Side panel for time graph
    conditionalPanel(
      # Check the  graph is selected
      condition = "input.tabs == 'use' && input.tabs_use == '3dpiece'",
      ns = NS(NULL),

      # Panel title
      h3("3D Distance Analysis by Piece"),
      # Context
      h5(dynamicDescription("3D Distance Analysis by Piece"))
    ),

    # Side panel for time graph
    conditionalPanel(
      # Check the  graph is selected
      condition = "input.tabs == 'use' && input.tabs_use == '3dplayer'",
      ns = NS(NULL),
```

```
    # Panel title
    h3("3D Distance Analysis by Player"),
    # Context
    h5(dynamicDescription("3D Distance Analysis by Player"))
   ),

   # Side panel for time graph
   conditionalPanel(
    # Check the  graph is selected
    condition = "input.tabs == 'heat' && input.tabs_heat == 'heat'",
    ns = NS(NULL),

    # Panel title
    h3("Player Positions Heatmap"),
    # Context
    h5(dynamicDescription("Player Positions Heatmap"))
   ),

   # Side panel for time graph
   conditionalPanel(
    # Check the  graph is selected
    condition = "input.tabs == 'heat' && input.tabs_heat == 'loc1'",
    ns = NS(NULL),

    # Panel title
    h3("Player Location Animation 1"),
    # Context
    h5(dynamicDescription("Player Location Animation 1"))
   ),

   # Side panel for time graph
   conditionalPanel(
    # Check the  graph is selected
    condition = "input.tabs == 'heat' && input.tabs_heat == 'loc2'",
    ns = NS(NULL),

    # Panel title
    h3("Player Location Animation 2"),
    # Context
    h5(dynamicDescription("Player Location Animation 2"))
   )
  ),

  mainPanel(
```

```
tabsetPanel(
  type = "tabs",
  # id is used for conditionals in side panels
  id = "tabs",
  tabPanel(
    # id and value used for conditionals in side panels
    id = "use",
    value = "use",

    # Panel title
    "User Engagement Visualizations",

    # Individual panels
    tabsetPanel(
      # id and value used for conditionals in side panels
      type = "tabs",
      id = "tabs_use",

      tabPanel(
        "Total Completion Time per Piece",
        plotlyOutput("completionTimePlot"),
        value = "completion"
      ),

      tabPanel(
        "Video Engagement Analysis",
        plotlyOutput("videoEngagementPlot"),
        value = "video"
      ),

      tabPanel(
        "3D Distance Analysis by Piece",
        plotlyOutput("distancePlot2"),
        value = "3dpiece"
      ),

      tabPanel(
        "3D Distance Analysis by Player",
        plotlyOutput("distancePlot1"),
        value = "3dplayer"
      )
    )
  ),
```

```
            tabPanel(
              # id and value used for conditionals in side panels
              id = "heat",
              value = "heat",

              # Panel title
              "Heatmap Visualizations",

              # Individual panels
              tabsetPanel(
                # id and value used for conditionals in side panels
                type = "tabs",
                id = "tabs_heat",
                tabPanel(
                  "Player Positions Heatmap",
                  plotlyOutput("heatmapPlot"),
                  value = "heat"
                ),

                tabPanel(
                  "Player Location Animation Draft 1",
                  imageOutput("locationAnimation1"),
                  value = "loc1"
                ),

                tabPanel(
                  "Player Location Animation Draft 2",
                  plotlyOutput("locationAnimation2"),
                  value = "loc2"
                )
              )
            )
          ))
        ),
      ),

      tabPanel(
        "ARCS Model Evaluation",
        htmlOutput("googleFormTab"))
    )
  )
)
```