```
library(shiny)

ui <- fluidPage(
  navbarPage("MAA",
             tabPanel("Data Set",
                      sidebarPanel(
                        fileInput("file", "Upload", multiple = FALSE,
                                  accept = c("text/csv",
                                             "text/comma-separated-values,text/plain",
                                             ".csv")),
                        tags$hr(),
                        radioButtons("vars", "Select data set:", choices = c("Full", "Columns")),
                        selectInput("cols", "Choose Variables:", choices = "", selected = "", multiple = TRUE)
                      ),
                      mainPanel(tableOutput("table"))
             ),

            navbarMenu("Descriptives",
                       tabPanel("Summary Tables",
                                sidebarPanel(
                                  selectInput("cols1", "Choose Variables:", choices = "", selected = "", multiple = 
TRUE)
                                ),
                                mainPanel(
                                  verbatimTextOutput("summar")
                                )
                        ),
                       tabPanel("Frequency Tables",
                                sidebarPanel(
                                  selectInput("cols2", "Choose Variable:", choices = "", multiple = TRUE),
                                  selectInput("cols3", "Choose Variable:", choices = "", multiple = TRUE)
                                         ),
                                mainPanel(
                                  verbatimTextOutput("freq_tab")
                                )
                       ),
                       tabPanel("Plots",
                                sidebarPanel(
                                  radioButtons("plotOpt", "Select Types:", choices = c("histogram", "bar", "scatter", 
"pie")),

                                  selectInput("cols4", "Choose Variable:", choices = " ", multiple = TRUE)
                                  ),
                                mainPanel(
                                  plotOutput("plot")
                                )
                       )
              ),
            navbarMenu("Marketing Analytics",
              tabPanel("Market Share",
                       sidebarLayout(
                         sidebarPanel(
                           selectInput("msvarinput", "Choose the variable:", choices = "", selected = ""),
                           textInput("msmetric", "Choose Metric:", placeholder = "Write the metric name"),
                           hr(),
                           helpText("Following metrics are supported:"),
                           tags$ol(
                             tags$li("Market Share (ms)"),
                             tags$li("Brand Development Index (bdi)")
                             ),
                           hr()

                           ),
                         mainPanel(
                           uiOutput("marketshare"),
                           div(
                             tableOutput("mshare")
                           )
                         )
                       )
                       )


            ),
            tabPanel("Contact",
                     sidebarPanel(
                       tags$h4("Contact")

                     ),
                     mainPanel(textOutput("text")))
  )

)

server <- function(input, output, session) {

  data <- reactive({
    req(input$file)
    data.frame(read.csv(input$file$datapath))
  })

  observeEvent(input$file, {
    updateSelectInput(session, inputId = "cols", choices = names(data()))
  })

  output$table <- renderTable(
    {
      df <- data()
      if(input$vars == "Full"){
        print(df)
      } else {
        print(df[input$cols])
      }
    }
  )

  observeEvent(input$file, {
    updateSelectInput(session, inputId = "cols1", choices = names(data()))
  })

  # summary tab

  summa <- reactive({
    var <- data()[, input$cols1]
    su <- summary(var)
    return(su)
  })

  output$summar <- renderPrint({
    summa()
  })

  # table tab

  observeEvent(input$file, {
    updateSelectInput(session, inputId = "cols2", choices = names(data()))
    updateSelectInput(session, inputId = "cols3", choices = names(data()))
  })

  tab <- reactive({
    var1 <- data()[, input$cols2]
    var2 <- data()[, input$cols3]
    ta <- table(var1, var2)
    return(ta)
  })

  output$freq_tab <- renderPrint({
    tab()
  })

  # plots

  observeEvent(input$file, {
    updateSelectInput(session, inputId = "cols4", choices = names(data()))

  })

  output$plot <- renderPlot({
    df <- data()

    if(input$plotOpt == "histogram"){
      hist(df[,input$cols4], freq = FALSE)
    } else if(input$plotOpt == "bar"){
      barplot(df[,input$cols4])
    } else if(input$plotOpt == "scatter"){
      plot(df[,input$cols4])
    } else {
      pie(table(df[,input$cols4]))
    }
  })
  observeEvent(input$file, {
    updateSelectInput(session, inputId = "msvarinput", choices = names(data()))

  })

  msout <- reactive({
    df <- data()
    len <- length(df[, input$msvarinput])
    total <- sum(df[, input$msvarinput])

    # df <- abs(round(rnorm(10)*10, 2))
    x <- matrix(NA, len, 1)
    for (i in 1:len){
      # x[i] <- df[, input$msvarinput][i]/total
      x[i] <- df[, input$msvarinput][i]/total
    }
    return(cbind.data.frame(company = df[, 1], market_share = x))


  })

  rmsout <- reactive({
    df <- data()
    len <- length(df[, input$msvarinput])
    total <- sum(df[, input$msvarinput])
    # df <- abs(round(rnorm(10)*10, 2))
    x <- matrix(NA, len, 1)
    for (i in 1:len){
      # x[i] <- df[, input$msvarinput][i]/total
      x[i] <- df[, input$msvarinput][i]/total
    }
    lcms <- max(x)
    return(cbind.data.frame(company = df[, 1], market_share = (x/lcms)))


  })

  output$marketshare <- renderUI({
    if (input$msmetric == "ms"){
      withMathJax(
        helpText("The Equation for 'Market Share': $$\\frac{Sales Revenue}{Total Market Revenue}$$")
      )
    } else if (input$msmetric == "rms"){
      withMathJax(
        helpText("Relative Market Share: $$\\frac{Brands Market Share}{Largest Competitors Marketshare}$$")
      )
    }
  }
  )

  output$mshare <- renderTable({
    if (input$msmetric == "ms"){
      msout()
    } else if(input$msmetric == "rms"){
      rmsout()
    }

  })

  #contct informtaion

  output$text <- renderText({
    "Contact +919177573730"

  }

  )
}

shinyApp(ui, server)
```