

Shiny Hands On Tutorial

Dewi Lestari Amaliah

Prerequisites

1. Having R and R studio installed. If you haven't already, you can see the installation guide [here](#).
2. Download the helper function [here](#). It is a function to calculate 10-year cardiovascular risk based on Framingham Risk Score.
3. Having these packages installed: `tidyverse`, `shiny`, `shinythemes`, `rmarkdown`, and `officer`. You can install these packages by running this code:

```
install.packages(c("tidyverse", "shiny", "shinythemes",  
                  "rmarkdown", "officer"))
```

Goal

Create a Shiny app to do health screening, i.e., BMI, blood sugar, blood pressure, and cardiovascular risk.

Your Task

Based on the lesson, think of the UI, server, and layout component of the app!

Steps

1. Create project: Open R Studio -> File -> New Directory -> Shiny Application
2. Start write the code for UI and Server

```
library(shiny)  
library(shinythemes)
```

```

library(officer)
library(tidyverse)

# Read helper function
source("helper.R")

# Define UI

ui <- navbarPage(
  theme = shinytheme("flatly"),
  "Health Screening Calculator",
  tabPanel(
    "Screening",
    # Sidebar layout with input and output definitions
    fluidRow(
      column(width = 3,
        textInput("patientName", "Patient Name:"),
        numericInput("age", "Age:", value = NULL, min = 1, max = 150),
        selectInput("sex", "Sex:", choices = c("Male", "Female")),
        numericInput("height", "Height (cm):", value = NULL, min = 50, max = 250, step = 10),
        numericInput("mass", "Mass (kg):", value = NULL, min = 1, max = 300, step = 10),
        selectInput("smoker", "Smoker:", choices = c("Yes", "No"))),
      column(width = 3,
        numericInput("systolic", "Systolic BP (mmHg):", value = NULL, min = 60, max = 180),
        numericInput("diastolic", "Diastolic BP (mmHg):", value = NULL, min = 40, max = 120),
        numericInput("bloodSugar", "Fasting Blood Sugar (mg/dL):", value = NULL, min = 50, max = 300),
        numericInput("totalChol", "Total Cholesterol (mg/dL):", value = NULL, min = 50, max = 300),
        numericInput("hdlChol", "HDL Cholesterol (mg/dL):", value = NULL, min = 10, max = 100),
        actionButton("calcStatus", "Calculate Status")),
      column(width = 6,
        h4(textOutput("t1")),
        textOutput("bmiResult"),
        textOutput("bmiClass"),
        br(),
        h4(textOutput("t2")),
        textOutput("bpClass"),
        br(),
        h4(textOutput("t3")),
        textOutput("bsClass"),
        br(),
        h4(textOutput("t4")),

```

```

        textOutput("cvRisk")
      )
    ),
    tabPanel(
      "Report",
      downloadButton("downloadPDF", "Download Health Report")
    )
  )

# Define the server logic
server <- function(input, output, session) {

  observeEvent(input$calcStatus, {

    # Ensure the inputs are valid
    if(is.null(input$height) || is.null(input$mass) || is.null(input$systolic) ||
       is.null(input$diastolic) || is.null(input$bloodSugar) || is.null(input$sex) ||
       is.null(input$smoker) || is.null(input$patientName) || is.null(input$age) ||
       is.null(input$totalChol) || is.null(input$hdlChol) || input$height <= 0 ||
       input$mass <= 0 || input$systolic <= 0 || input$diastolic <= 0 ||
       input$bloodSugar <= 0 || input$patientName == "" || input$age <= 0 ||
       input$totalChol <= 0 || input$hdlChol <= 0) {
      output$bmiResult <- renderText("Please enter all required information.")
      output$bmiClass <- renderText("")
      output$bpClass <- renderText("")
      output$bsClass <- renderText("")
      output$cvRisk <- renderText("")
    } else {

      # Calculate BMI
      height_m <- input$height / 100
      bmi <- input$mass / (height_m ^ 2)

      # Determine the BMI classification
      bmiClass <- ifelse(bmi < 18.5, "Underweight",
                        ifelse(bmi < 24.9, "Normal weight",
                              ifelse(bmi < 29.9, "Overweight",
                                    ifelse(bmi < 34.9, "Obesity I",
                                            ifelse(bmi < 39.9, "Obesity II", "Obesity III"))))
    }
  })
}

```

```

# Determine the blood pressure classification
systolic <- input$systolic
diastolic <- input$diastolic
bpClass <- ifelse(systolic < 120 & diastolic < 80, "Normal",
                  ifelse(systolic < 130 & diastolic < 80, "Elevated",
                        ifelse(systolic < 140 | diastolic < 90, "Hypertension Stage 1", "Hypertension Stage 2")))

# Determine the blood sugar classification
bloodSugar <- input$bloodSugar
bsClass <- ifelse(bloodSugar < 100, "Normal",
                  ifelse(bloodSugar < 126, "Prediabetes", "Diabetes"))

# Calculate Cardiovascular Risk
cvRisk <- calculate_framingham_risk(input$age, input$sex,
                                   input$totalChol,
                                   input$hdlChol,
                                   input$systolic,
                                   input$smoker)

# Render the results
output$t1 <- renderText("Body Mass Index (BMI) Status")
output$t2 <- renderText("Blood Pressure Status")
output$t3 <- renderText("Blood Sugar Status")
output$t4 <- renderText("Cardiovascular Risk")
output$bmiResult <- renderText(paste("Your BMI is", round(bmi, 1)))
output$bmiClass <- renderText(paste("BMI Classification:", bmiClass))
output$bpClass <- renderText(paste("Blood Pressure Classification:", bpClass))
output$bsClass <- renderText(paste("Blood Sugar Classification:", bsClass))
output$cvRisk <- renderText(paste("Cardiovascular Risk:", cvRisk))

# Download PDF
output$downloadPDF <- downloadHandler(
  filename = function() {
    paste(input$patientName, "_health_report", ".pdf", sep = "")
  },
  content = function(file) {
    # Create a temporary R Markdown file
    report <- tempfile(fileext = ".Rmd")
    on.exit(unlink(report))

    # Write the content

```

```

# Write the content of the R Markdown file
writeLines(c(
  "----",
  "title: 'Health Report'",
  "output: pdf_document",
  "----",
  "",
  paste("# Health Report for", input$patientName),
  paste("## Age:", input$age),
  paste("## Sex:", input$sex),
  "",
  paste("## BMI Status:",
  paste("* Your BMI is", round(bmi, 1)),
  paste("* BMI Classification:", bmiClass),
  "",
  paste("## Blood Pressure Status:",
  paste("* Blood Pressure Classification:", bpClass),
  "",
  paste("## Blood Sugar Status:",
  paste("* Blood Sugar Classification:", bsClass),
  "",
  paste("## Cardiovascular Risk:",
  paste("* Cardiovascular Risk:", cvRisk)
), report)

# Render the R Markdown file to PDF
rmarkdown::render(report, output_file = file, envir = new.env())
}
)
}
})
}

# Run the application
shinyApp(ui = ui, server = server,
  options = list(launch.browser = TRUE))

```