Supplementary

Appendix 1

```
#Supplemental File 1: Shiny R file 'server.R'
library(shiny)
library(shinydashboard)
library(glmnet)

#load model .rds packaged with shiny app deployment
mort_model <- readRDS('./tutorial_model.rds')

#alternate .rds file storage option; upload to github repository of choice and reference the download
link as follows:
#mort_model <- readRDS(gzcon(url("https://github.com/S-
SPIRE/clinical_calculators/raw/main/tutorial_model.rds")))

shinyServer(function(input, output, session){
 #reactive function assigns user input to variable table used in predictions
 inputData <- reactive({
  female <- 0

  age_74 <- 0
  age_79 <- 0
  age_84 <- 0
  age_89 <- 0
  age_90 <- 0

  fnstatus_partd <- 0
  fnstatus_totd <- 0

  dyspnea_atrest <- 0
  dyspnea_modexe <- 0

  prsepis_sirs <- 0
  prsepis_sepsis <- 0
  prsepis_shock <- 0

  if (input$Sex == "Female"){
   female <- 1
  }


  if (input$Age < 75 & input$Age >= 74){
   age_74 <- 1
  } else if (input$Age < 80 & input$Age >= 75){
   age_79 <- 1
  } else if (input$Age < 85 & input$Age >= 80){
   age_84 <- 1
  } else if (input$Age < 90 & input$Age >= 85){
   age_89 <- 1
  } else if (input$Age >= 90){
```


https://dx.doi.org/10.21037/atm-22-847

```
    age_90 <- 1
   }

   if (input$healthStatus == "Partially Dependent") {
    fnstatus_partd <- 1
   } else if (input$healthStatus == "Totally Dependent") {
    fnstatus_totd <- 1
   }

   if (input$Dyspnea == "At rest") {
    dyspnea_atrest <- 1
   } else if (input$Dyspnea == "Moderate exertion") {
    dyspnea_modexe <- 1
   }

   if (input$Sepsis == "SIRS") {
    prsepis_sirs <- 1
   } else if (input$Sepsis == "Sepsis") {
    prsepis_sepsis <- 1
   } else if (input$Sepsis == "Septic Shock") {
    prsepis_shock <- 1
   }

   #function returns a single row data frame containing variables needed for model prediction
   #variable names in this data frame must match those of the original dataset used to train the model
   userPredictData <- data.frame("female" = female, "age_74" = age_74, "age_79" = age_79, "age_84" =
age_84, "age_89" = age_89, "age_90" = age_90, "fnstatus_partd" = fnstatus_partd, "fnstatus_totd" =
fnstatus_totd, "dyspnea_atrest" = dyspnea_atrest, "dyspnea_modexe" = dyspnea_modexe,
"prsepis_sirs" = prsepis_sirs, "prsepis_sepsis" = prsepis_sepsis, "prsepis_shock" = prsepis_shock)
  })

  #function takes an as argument the imported model object and returns the model prediction based on
data returned by inputData()
  generate_model <- function(m){
   dat <- as.matrix(inputData())
   p <- predict(m, newx = dat, type = "response", s = "lambda.1se")
   p
  }

  #render gauge with predicted value for display in ui.R
  output$mortality_gauge <-
flexdashboard::renderGauge(flexdashboard::gauge(round(generate_model(mort_model)*100, 1),
symbol = "%",min = 0, max = 100, sectors = gaugeSectors(success = c(0, 5), warning = c(5, 5*2), danger =
c(5*2, 100)), abbreviate = TRUE, abbreviateDecimals = 0))

  #render display box with predicted value for display in ui.R
```


https://dx.doi.org/10.21037/atm-22-847

```
  output$death_box <- renderInfoBox({infoBox("Mortality Risk", value = tags$p(style = "font-size:
35px;",paste0(round(generate_model(mort_model)*100, 1), "%")),  icon = icon("skull"), color = "red", fill
= TRUE)})
})
```


https://dx.doi.org/10.21037/atm-22-847

Appendix 2

```
#Supplementary File 2: Shiny R file 'ui.R'
library(shiny)
library(shinythemes)
library(shinydashboard)
library(flexdashboard)
library(htmltools)

#lists and creates navigational features in sidebar menu
menu_sidebar <- dashboardSidebar(
  sidebarMenu(id = "tabs",
          menuItem("Calculator Inputs", tabName = "predictors", icon = icon("calculator")),
          menuItem("Calculator Results", tabName = "risk", icon = icon("gauge-high"))
  )
)

#input_box and display_gauge contain application features organized into task-relevant components
input_box <- box(title ="Patient characteristics", status = "danger", header = T, solidHeader = T,  width =
12,
             numericInput("Age", "Age (yrs)", 50, min = 0, max = 120),
             selectInput("Sex", "Sex", choices = c("Male", "Female")),
             selectInput("Sepsis", "Systemic sepsis", choices = c("None", "SIRS", "Sepsis", "Septic Shock")),
             selectInput("Dyspnea", "Dyspnea", choices = c("No", "Moderate exertion", "At rest")),
             selectInput("healthStatus", "Function health status prior to surgery", choices =
c("Independent", "Partially Dependent", "Totally Dependent")))

display_gauge <- box(title = "MORTALITY RISK", align = "center", status = "primary", solidHeader = TRUE,
flexdashboard::gaugeOutput("mortality_gauge"))

#body takes the components created in previous items and assigns them to proper tab windows
body <- dashboardBody(tabItems(
  tabItem(tabName = "predictors", input_box),
  tabItem(tabName = "risk", display_gauge, infoBoxOutput("death_box", width = 6))
))

#dashboardPage combines all application content including the sidebar and body into one coherent UI
dashboardPage(
  dashboardHeader(title = span("Tutorial: Risk Calculator", style = "font-size:14px"),  titleWidth = 300),

  menu_sidebar,
  body
)
```


https://dx.doi.org/10.21037/atm-22-847