myApp <- function(...){

library(shinyjs)

library(shiny)

library(shinydashboard)

library(shinydashboardPlus)

library(echarts4r)

dat1 <- read.csv("Harbin.csv")

dat1$DATE <- as.Date(dat1$DATE)

# 定义UI

ui <- dashboardPage(

title = "积温计算器",

dashboardHeader(title = "积温计算器"),

dashboardSidebar(

sidebarMenu(

menuItem("使用说明", tabName = "info", icon = icon("info-circle")),

menuItem("积温计算", tabName = "tempCalculation", icon = icon("calculator"))

)

),

dashboardBody(

useShinyjs(),

tags$head(

tags$style(HTML('

/\* 欢迎区域样式 \*/

.welcome-section {

background-color: #3C8DBC;

color: #FFFFFF;

padding: 30px;

border-radius: 10px;

box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);

margin: 20px;

opacity: 0;

transition: opacity 1s;

}

/\* 使用说明区域样式 \*/

.info-section {

background-color: #6BB9F0;

padding: 30px;

border-radius: 10px;

box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);

margin: 20px;

}

/\* 注意事项区域样式 \*/

.notice-section {

background-color: #FF5733;

color: #FFFFFF;

padding: 30px;

border-radius: 10px;

box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);

margin: 20px;

}

/\* 标题样式 \*/

h4 {

font-size: 24px;

text-transform: uppercase;

}

/\* 段落样式 \*/

p {

font-size: 18px;

line-height: 1.5;

}

/\* 按钮样式 \*/

.action-button {

background-color: #3C8DBC;

color: #FFFFFF;

border: none;

border-radius: 5px;

padding: 10px 20px;

font-size: 18px;

transition: background-color 0.3s;

}

.action-button:hover {

background-color: #3071A9;

}

/\* 底部样式 \*/

.custom-footer {

margin-left: -10px;

background-color: #3C8DBC;

color: #FFFFFF;

padding: 10px;

text-align: center;

border-radius: 5px;

margin-top: 20px;

}

/\* 自定义输入框样式 \*/

.custom-input input[type="text"],

.custom-input input[type="date"] {

background-color: #F0F0F0;

border: 1px solid #CCCCCC;

border-radius: 5px;

padding: 5px 10px;

font-size: 16px;

width: 100%;

}

'))

),

tabItems(

tabItem(tabName = "info",

fluidRow(

column(

width = 12,

tags$div(

class = "welcome-section",

h4("欢迎使用积温计算器！"),

p("本计算器提供了一种简便的方式来计算作物生长期间的累积温度。"),

p("请按照以下步骤使用本计算器："),

HTML("<ol>

<li>输入您的播种日期和收获日期。</li>

<li>输入作物零点温度°C。</li>

<li>点击“确定”按钮以获取积温计算结果。</li>

</ol>")

),

tags$script(HTML('

shinyjs.animateWelcome = function() {

$(".welcome-section").css("opacity", "1");

}

$(document).ready(function() {

shinyjs.animateWelcome();

});

'))

),

column(

width = 12,

tags$div(

class = "info-section",

h4("使用说明"),

p("本积温计算器提供了一种简便的方式来计算作物生长期间的累积温度。"),

p("积温是一个农业气象学的概念，指作物生长期间的累积有效温度总和。"),

p("计算结果包括从播种到收获期间的总天数、大于作物生长下限温度的天数、最长连续符合生长条件的天数，以及相应的积温和有效积温。"),

p("有效积温是指在生物学意义上对作物生长有益的温度累积。")

)

),

column(

width = 12,

tags$div(

class = "notice-section",

h4("注意事项"),

p("积温计算结果仅供参考，实际作物生长情况可能受多种因素影响。"),

p("建议结合当地气候条件和农业实践经验进行综合判断。")

)

)

),

tags$footer(

class = "custom-footer",

"感谢您使用积温计算器 © 2023"

)

),

tabItem(tabName = "tempCalculation",

fluidRow(

box(

title = "积温计算条件",

status = "primary",

solidHeader = TRUE,

collapsible = FALSE,

width = 6,

div(

class = "custom-input",

dateInput("start.date", "开始播种日期", value = "2012-09-18")

),

div(

class = "custom-input",

dateInput("end.date", "收获日期", value = "2013-09-17")

),

div(

class = "custom-input",

numericInput("obs", "作物零点温度°C:", 10, min = 0, max = 20)

),

actionButton("done", "确定", class = "btn btn-primary action-button"),

p("温度数据只能提供2010.10-2023.11")

),

box(

"Box body",

id = "mybox",

collapsible = TRUE,

closable = TRUE,

textOutput("daysmax"),

textOutput("continous\_day"),

textOutput("accumulated\_t"),

textOutput("effective\_a\_t"),

textOutput("day\_duration"),

echarts4rOutput("plot")

)

),

tags$footer(

class = "custom-footer",

"感谢您使用积温计算器 © 2023"

)

)

)

)

)

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

observeEvent(input$done, {

date\_range <- reactive(subset(dat1,

DATE>=input$start.date&

DATE<=input$end.date))

date\_range <- isolate(date\_range())

plant\_zero <- isolate(input$obs)

days <- which(date\_range$TAVG>=plant\_zero)

day\_count <- length(days )

output$day\_duration<- renderText(

paste("选择时期历时:",nrow(date\_range),"天")

)

output$daysmax <- renderText(

paste("大于生物学零点的总数为:",day\_count,"天")

)

max\_day <- max\_day\_rec <- 1

for(i in 2:day\_count)

{

if(days[i]==days[(i-1)]+1)

{

max\_day <- max\_day+1

start\_p <- i-max\_day+1

}

else if(max\_day!=1)

{

if(max\_day>max\_day\_rec)

{

max\_day\_rec<- max\_day

start\_p\_rec <- start\_p

max\_day <- 0

}

}

}

# output2 maximum consecutive days

final\_day <- max(max\_day,max\_day\_rec)

output$continous\_day <- renderText(

paste("大于生物学零点的最长连续天数为:",final\_day,"天")

)

start\_point <- ifelse(max\_day>max\_day\_rec,

start\_p,

start\_p\_rec)

start\_point\_date <- date\_range$DATE[days[start\_point]]

end\_point\_date <- date\_range$DATE[days[(start\_point+final\_day-1)]]

ac\_temp <- sum(date\_range$TAVG[

date\_range$DATE>=start\_point\_date&date\_range$

DATE<=end\_point\_date])

output$accumulated\_t <- renderText(

paste("积温:",ac\_temp,"°C")

)

effective\_a\_t <- ac\_temp-final\_day\*plant\_zero

output$effective\_a\_t <- renderText(

paste("有效积温:", effective\_a\_t,"°C")

)

output$plot <- renderEcharts4r({

req(!input$mybox$collapsed)

yearMonthDate <- htmlwidgets::JS('function (value) {

var monthShortNames = ["一月", "二月", "三月", "四月", "五月", "六月",

"七月", "八月", "九月", "十月", "十一月", "十二月"

];

var d = new Date(value);

var datestring = monthShortNames[d.getMonth()] + " "

var datestring = d.getFullYear() + " " + monthShortNames[d.getMonth()] + " "

return datestring

}')

avg <- list(

type = "average",

name = "AVG"

)

date\_range|>

e\_charts(DATE)|>

e\_line(TAVG)|>

e\_mark\_area('TAVG',data = list(

list(xAxis = start\_point\_date,yAxis=plant\_zero),

list(xAxis = end\_point\_date,yAxis=max(dat1$TAVG),

title = "最大连续积温区域")

),

itemStyle = list(color = "lightgreen"))|>

e\_mark\_line(data = list(yAxis = plant\_zero),

title = "生物学下限温度",

itemStyle = list(color = "red")) |>

e\_format\_y\_axis(suffix = "°C")|>

e\_axis\_labels(

x = "日期",

y = "日均气温"

)|>

e\_x\_axis(

DATE,

axisPointer = list(show = TRUE),

axisLabel = list(

rotate = 45,

formatter = yearMonthDate

))|>

e\_tooltip(trigger = "axis")|>

e\_datazoom(

type = "slider",

toolbox = FALSE,

bottom = -5

)

})

output$box\_state <- renderText({

state <- ifelse(input$mybox$collapsed,"collapsed", "uncollapsed")

paste("My box is", state)

})

})

}

shinyApp(ui, server)

}