**Shiny Project:**

By: Girija

**Introduction:**

Shiny is a one of the great and beautiful packages in R Shiny is a package in R that allows you to create dynamic interactive webpages using relatively simple R statements.

Our objective here is to demonstrate the shiny package. It consists of both UI and server R code. We need to create a folder in which all the required data including ppts pdfs etc to be published on the page including the ui and server r codes should be available.

My intension is to create a shiny project with all data science courses and whatever we have learned in this Coursera course. My web page consists of all the ppts, pdfs, code docs etc.

**Ui code for the same:**

rm(list = ls(all = TRUE))

library(shiny)

shinyUI(fluidPage(

titlePanel("Project Shiny"),

sidebarLayout(

sidebarPanel(list(HTML('<style type="text/css">

.span4 .well { background-color: #B0E0E6;width:250px ; color :#F70D1A;}

</style>'),

navlistPanel(id="id1",selected="select",tabPanel("select"),

navbarMenu('Data Science Toolkit',tabPanel("Data"),

tabPanel("Fundamentals")),

navbarMenu('R Programming',tabPanel("About R"),

tabPanel("Programming")),

navbarMenu('Statistical Inference',tabPanel("Formulas"),

tabPanel("Inference")),

navbarMenu('Regression Models',tabPanel("Regression"),

tabPanel("Models")),

navbarMenu('Getting and Cleaning data',tabPanel("Loading Data"),

tabPanel("Cleaning data"),

tabPanel("Data munging")),

navbarMenu('Practical Machine Learning',tabPanel("About ML"),

tabPanel("Algorithms"))),

img(src="image/16170603-social-media-concept--abstract-illustration-with-hexagons-and-icons.png",height=100,width=100)

)),

mainPanel(list(HTML('<style type="text/css">

.span8 .well { background-color: #987654;fontzzcolor:#F70D1A;}

</style>'),

h4("main panel"),

conditionalPanel(

condition = "input.id1 == 'select'",

wellPanel("welcome",verbatimTextOutput("welcome"),img(src="image/ana.jpg",height = 500, width = 900))

),

conditionalPanel( condition="input.id1 != 'select'",

tabsetPanel(type="tabs",

tabPanel("Overview",

sidebarPanel("Data",verbatimTextOutput("Data"),width=10),

sidebarPanel("Process",verbatimTextOutput("Process"),width=5),

sidebarPanel("Models",verbatimTextOutput("Models"),width=5)),

tabPanel("Data",wellPanel(

fluidRow( column(4,

imageOutput("Data",width="350%",height="300%",inline=FALSE))))),

tabPanel("CodeandProcess",sidebarPanel("CodeandProcess",verbatimTextOutput("CodeandProcess"))),

tabPanel("Plots",wellPanel(

fluidRow( column(4,

imageOutput("Plots",width="350%",height="300%",inline=FALSE)))))

)),

conditionalPanel(

condition = "input.id1 == 'DataScienceToolkit '",

helpText ( a ("ppt",href="DataScienceToolkit.pdf",target="\_blank")),

helpText( a ("Doc",href="DataScienceToolkitDoc.pdf",target="\_blank"))

),

conditionalPanel( condition = "input.id1 == 'RProgramming'",

helpText ( a ("ppt",href="RProgramming.pdf",target="\_blank")),

helpText( a ("Doc",href="RProgrammingDoc.pdf",target="\_blank"))

),

conditionalPanel( condition = "input.id1 == 'StatisticalInference'",

helpText ( a ("ppt",href="StatisticalInference.pdf",target="\_blank")),

helpText( a ("Doc",href="StatisticalInferenceDoc.pdf",target="\_blank"))

),

conditionalPanel( condition = "input.id1 == 'RegressionModels'",

helpText ( a ("ppt",href="RegressionModels.pdf",target="\_blank")),

helpText( a ("Doc",href="RegressionModelsDoc.pdf",target="\_blank"))

),

conditionalPanel( condition = "input.id1 == 'GettingandCleaning data'",

helpText ( a ("ppt",href="GettingandCleaning data.pdf",target="\_blank")),

helpText( a ("Doc",href="GettingandCleaning dataDoc.pdf",target="\_blank"))

),

conditionalPanel( condition = "input.id1 == 'PracticalMachineLearning'",

helpText ( a ("ppt",href="PracticalMachineLearning.pdf",target="\_blank")),

helpText( a ("Doc",href="PracticalMachineLearningDoc.pdf",target="\_blank"))

),

conditionalPanel( condition = "input.id1 == 'ExploratoryDataAnalysis'",

helpText ( a ("ppt",href="ExploratoryDataAnalysis.pdf",target="\_blank")),

helpText( a ("Doc",href="ExploratoryDataAnalysisDoc.pdf",target="\_blank"))

),

conditionalPanel(

condition = "input.id1 == 'ReproducibleResearch'",

helpText ( a ("ppt",href="ReproducibleResearch.pdf",target="\_blank")),

helpText( a ("Doc",href="ReproducibleResearchDoc.pdf",target="\_blank"))

),

conditionalPanel(

condition = "input.id1 == 'DevelopingDataProducts'",

helpText ( a ("ppt",href="DevelopingDataProducts.pdf",target="\_blank")),

helpText( a ("Doc",href="DevelopingDataProductsDoc.pdf",target="\_blank"))

),

conditionalPanel(

condition = "input.id1 == 'Shiny'",

helpText ( a ("ppt",href="Shiny.pdf",target="\_blank")),

helpText( a ("Doc",href="ShinyDoc.pdf",target="\_blank"))

),

conditionalPanel(

condition = "input.id1 == 'swirl'",

helpText ( a ("ppt",href="swirl.pdf",target="\_blank")),

helpText( a ("Doc",href="swirlDoc.pdf",target="\_blank"))

),

tabsetPanel(tabPanel("Help",htmlOutput('help')))

))

))

)

**Server code :**

rm(list = ls(all = TRUE))

library(shiny)

library("png")

filenames <- dir("C:\\Users\\sai\\Desktop\shiny\\www\\","\*.pdf")

value <- read.csv("C:\\Users\\sai\\Desktop\\shiny\\Process.csv",header=T,sep="\t")

print(value$label)

i <- sapply(value,is.factor)

value[i] <- lapply(value[i],as.character)

shinyServer(function(input,output){

panelInput <- reactive({

switch(input$id1,"1" = DataScienceToolkit,"2" = RProgramming,"3" = SttsticalInference,"4" = RegressionModels,"5" = GettingandCleaningData,

"6" = PracticalMachineLearning,"7" = ExploratoryDataAnalysis,"8" = ReproducibleResearch,"9" =

DevelopingDataProducts,"10"=Shiny,"11"=swirl)

})

output$welcome <- renderPrint({

print("Now letâs define a skeletal server implementation. To do this we call shinyServer and pass it a function that accepts two parameters: input and output:Our server function is empty for now but later weâll use it to define the relationship between our inputs and outputs.Weâve now created the most minimal possible Shiny application. You can run the application by calling the runApp function as follows:")

})

output$Process<-renderText({

if(input$id1 %in% value$label){

t<-grep(input$id1,value$label,value=F,ignore.case=T)

tex<-value$process[t]

print(value$process[t])

}

else if(!(input$id1 %in% value$label)){

print("yet to know")

}

})

output$Technology<-renderText({

if(input$id1 %in% value$label){

t<-grep(input$id1,value$label,value=F,ignore.case=T)

tex<-value$technology[t]

print(value$technology[t])

}

else if(!(input$id1 %in% value$label)){

print("yet to be know")

}

})

output$Objective<-renderText({

if(input$id1 %in% value$label){

t<-grep(input$id1,value$label,value=F,ignore.case=T)

tex<-value$objective[t]

print(value$objective[t])

}

else if(!(input$id1 %in% value$label)){

print("yet to be know")

}

})

fun <- function(){ return(input$id1)}

output$site <- renderUI({

if(input$id1 %in% value$label){

HTML('<script language="javascript" type="text/javascript">

var scrt\_var = input$id1;

alert(scrt\_var)

</script>

<a href="" onclick="location.href=scrt\_var;return true;">Link</a>

')

}

})

output$Results <-renderImage({

val2 <- input$id1

val1 <-"C:\\Users\\sai\\Desktop\\shiny\\www\\image\\"

val3 <-".png"

print(val2)

if(input$id1 %in% value$label){

return(list(

val <- paste(val1,val2,val3,sep=""),

src =val,

contentType = "image/png",

alt = "Face"

))

}

else {

if(!(input$id1 %in% value$label)){

return(list(

src ="C:\\Users\\sai\\Desktop\\shiny\\www\\image\\image.png",

contentType = "image/png",

alt = "Image"

))

}}

} , deleteFile = FALSE)

output$Benefits <-renderText({

if(input$id1 %in% value$label){

t<-grep(input$id1,value$label,value=F,ignore.case=T)

tex<-value$benefits[t]

print(value$benefits[t])

}

else if(!(input$id1 %in% value$label)){

print("yet to be know")

}

})

output$report = downloadHandler(

filename = paste("C:\\Users\\sai\\Desktop\\shiny\\www\\",input$id1,".pdf",sep=""),

content = function(file) {

if(input$id1 %in% value$label){

file.show(paste("C:\\Users\\sai\\Desktop\\shiny\\www\\",input$id1,".pdf",sep=""), file)}

},

contentType = 'application/pdf'

)

output$Plots <-renderImage({

val2 <- input$id1

val1 <-"C:\\Users\\sai\\Desktop\\shiny\\www\\image\\"

val3 <-".pdf"

print(val2)

if(input$id1 %in% value$label){

return(list(

val <- paste(val1,val2,val3,sep=""),

src =val,

contentType = "application/pdf",

alt = "Face"

))

}

else {

if(!(input$id1 %in% value$label)){

return(list(

src ="C:\\Users\\sai\\Desktop\\shiny\\www\\image\\image.png",

contentType = "image/png",

alt = "Image"

))

}}

} , deleteFile = FALSE)

output$help <- renderUI({

val <- paste("C:\\Users\\sai\\Desktop\\shiny\\www\\image\\",input$id1,".pdf",sep="")

print(val)

HTML('<iframe

seamless=NA,

src=val>')

})

})

\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*