library(shiny)

library(dplyr)

library(ggplot2)

library(plotly)

library(DT)

# Data Collection Module – Imports attendance data from CSV files

ui <- fluidPage(

titlePanel("Interactive Student Records Management"),

sidebarLayout(

sidebarPanel(

fileInput("file", "Upload CSV File", accept = ".csv"),

helpText("Ensure the CSV file has 'Student\_ID', 'Name', 'Class', 'Section', 'Date', and 'Attendance' columns."),

actionButton("process", "Process Data")

),

mainPanel(

tabsetPanel(

tabPanel("Student Summary", DTOutput("data\_summary")),

tabPanel("Class-wise Distribution", plotlyOutput("class\_distribution")),

tabPanel("Attendance Trends", plotlyOutput("attendance\_trend")),

tabPanel("Daily Attendance", plotlyOutput("daily\_bar"))

)

)

)

)

server <- function(input, output) {

# Data Cleaning Module – Handles missing values, removes duplicates, and corrects inconsistencies

student\_data <- reactive({

req(input$file)

df <- read.csv(input$file$datapath, stringsAsFactors = FALSE)

if (!all(c("Student\_ID", "Name", "Class", "Section", "Date", "Attendance") %in% colnames(df))) {

stop("Error: Missing required columns.")

}

df$Date <- as.Date(df$Date, format="%d/%m/%Y")

df$Attendance <- tolower(df$Attendance)

df$Attendance[df$Attendance %in% c("p", "pres", "present")] <- "Present"

df$Attendance[df$Attendance %in% c("a", "abs", "absent")] <- "Absent"

df[is.na(df)] <- "Absent"

df <- df[!duplicated(df), ]

df

})

# Data Standardization Module – Formats dates, normalizes attendance status, and standardizes student names

output$data\_summary <- renderDT({

req(input$process)

df <- student\_data()

datatable(df)

})

# Data Analysis & Reporting Module – Summarizes attendance trends

output$class\_distribution <- renderPlotly({

req(input$process)

df <- student\_data()

summary\_table <- df %>% group\_by(Class) %>% summarise(Count = n())

plot <- ggplot(summary\_table, aes(x = Class, y = Count, fill = Class)) +

geom\_bar(stat = "identity") +

labs(title = "Class-wise Student Distribution", x = "Class", y = "Count") +

theme\_minimal()

ggplotly(plot)

})

# Visualization Module – Uses ggplot2 to create graphs for attendance trends

output$attendance\_trend <- renderPlotly({

req(input$process)

df <- student\_data()

df$Attendance <- as.factor(df$Attendance)

plot <- ggplot(df, aes(x=Date, fill=Attendance)) +

geom\_bar(position="stack") +

labs(title="Attendance Trends", x="Date", y="Count") +

theme\_minimal()

ggplotly(plot)

})

output$daily\_bar <- renderPlotly({

req(input$process)

df <- student\_data()

df$Attendance <- as.factor(df$Attendance)

plot <- ggplot(df, aes(x=Date, fill=Attendance)) +

geom\_bar(position="dodge") +

labs(title="Daily Attendance Count", x="Date", y="Count") +

theme\_minimal()

ggplotly(plot)

})

}

# Run the Shiny app

shinyApp(ui = ui, server = server)