Take home Exercise 3

Yang Lu

2025-06-15

# **Take-home Exercise 3: Prototyping Modules for Visual Analytics Shiny Application**

# **Storyboard**

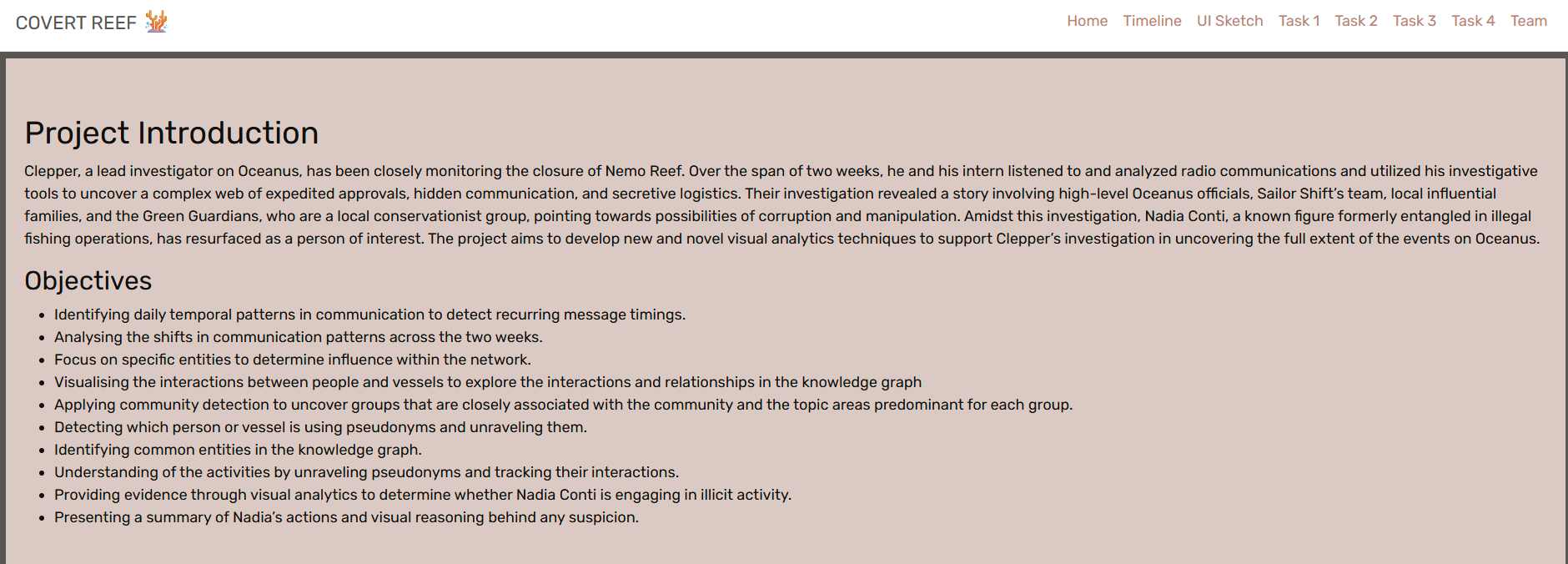
## Section 1: Design for UI theme +Navbar

There will split the section into : Homepage about project introduction, Project timeline, the hand draw UI sketch, Task 1, Task 2, Task 3 , Task 4 and Team member.

# --------------------------------------------------  
# 1. DIY bslib theme  
# --------------------------------------------------  
theme\_covert <- bs\_theme(  
 bg = "#585651",  
 fg = "#090605",  
 primary = "#ba9b8d",  
 secondary = "#814a38",  
 base\_font = font\_google("Rubik"),  
 heading\_font = font\_google("Rubik")  
)  
  
# --------------------------------------------------  
# UI  
# --------------------------------------------------  
ui <- fluidPage(  
 theme = theme\_covert,  
 tags$head(  
 tags$style(HTML("  
 html, body { scroll-behavior: smooth; }  
 .navbar-covert { background-color: #FFF; }  
 .navbar-covert .nav-link { color: #c17e72 !important; font-weight: 500; }  
 .navbar-covert .nav-link:hover { color: #DBCAC3 !important; }  
 body { padding-top: 70px; }  
 section { padding: 60px 20px; }  
 /\* web BG set for #DBCAC3 section \*/  
#home,  
#sketch,  
#task2,  
#task4 {  
 background-color: #DBCAC3 !important;  
}  
  
/\* web BG set for #ebe2de section \*/  
#timeline,  
#task1,  
#task3,  
#team {  
 background-color: #ebe2de !important;  
}  
  
 /\* TabsetPanel \*/  
 .nav-tabs .nav-link {  
 color: #000 !important;  
 background-color: #bda59f !important;  
 }  
 .nav-tabs .nav-item.show .nav-link,  
 .nav-tabs .nav-link.active {  
 color: #000 !important;  
 background-color: #bda59f !important;  
 }  
  
 /\* Dropdown box color matching \*/  
 .form-control.shiny-bound-input,  
 .selectize-control.single .selectize-input {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 border-color: #bda59f !important;  
 }  
 .selectize-dropdown {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 }  
  
 /\* dateRangeInput bg & text \*/  
 #date\_range .form-control {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 border-color: #bda59f !important;  
 }  
 #date\_range .input-group-text {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 border-color: #bda59f !important;  
 }  
 "))  
 ),  
   
 # Navbar  
 tags$nav(class = "navbar navbar-expand-lg navbar-covert fixed-top",  
 tags$div(class = "container-fluid",  
 tags$a(class = "navbar-brand text-white", href = "#home",  
 tags$span("COVERT REEF "), tags$span("\U0001FAB8", style = "font-size:24px;")  
 ),  
 tags$button(class = "navbar-toggler", type = "button",  
 `data-bs-toggle` = "collapse", `data-bs-target` = "#menuCovert",  
 tags$span(class = "navbar-toggler-icon")  
 ),  
 tags$div(class = "collapse navbar-collapse", id = "menuCovert",  
 tags$ul(class = "navbar-nav ms-auto",  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#home", "Home")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#timeline","Timeline")),  
 tags$li(class= "nav-item", tags$a(class="nav-link", href="#sketch", "UI Sketch")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task1", "Task 1")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task2", "Task 2")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task3", "Task 3")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task4", "Task 4")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#team", "Team"))  
 )  
 )  
 )  
 )

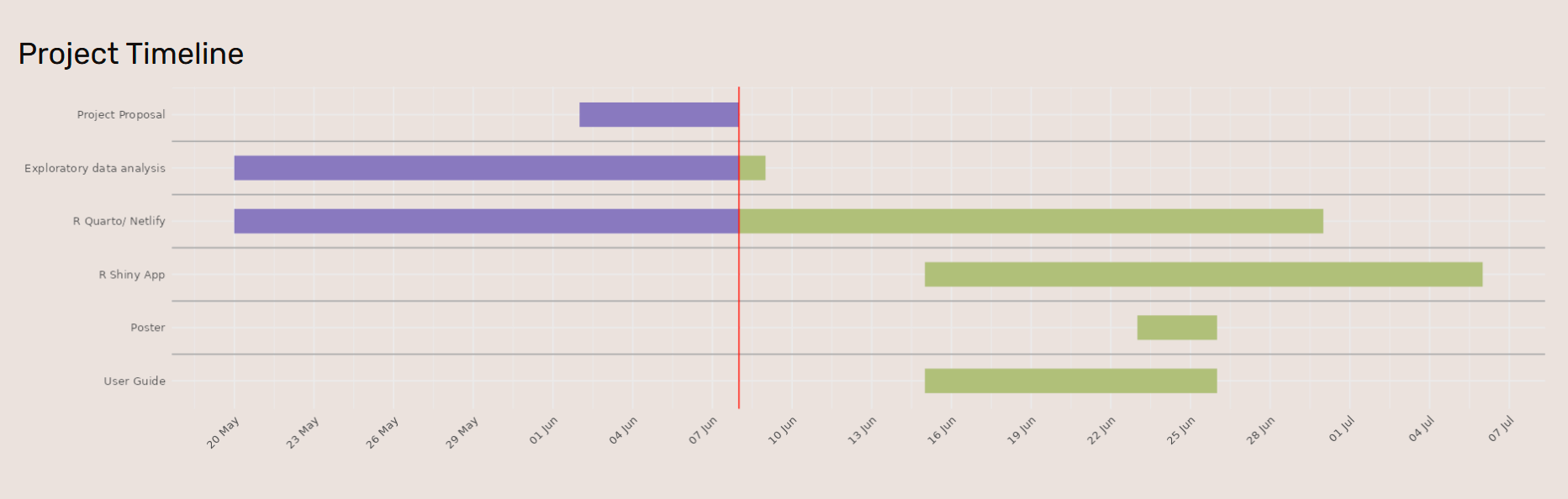
## Section 2 :Project Introduction

# Home Section  
 tags$section(id="home", fluidRow(column(12,  
 h1("Project Introduction"),  
 p("Clepper, a lead investigator on Oceanus, has been closely monitoring the closure of Nemo Reef. Over the span of two weeks, he and his intern listened to and analyzed radio communications and utilized his investigative tools to uncover a complex web of expedited approvals, hidden communication, and secretive logistics. Their investigation revealed a story involving high-level Oceanus officials, Sailor Shift’s team, local influential families, and the Green Guardians, who are a local conservationist group, pointing towards possibilities of corruption and manipulation. Amidst this investigation, Nadia Conti, a known figure formerly entangled in illegal fishing operations, has resurfaced as a person of interest. The project aims to develop new and novel visual analytics techniques to support Clepper’s investigation in uncovering the full extent of the events on Oceanus."),  
 h3("Objectives"),  
 tags$ul(  
 tags$li("Identifying daily temporal patterns in communication to detect recurring message timings.  
  
"),  
 tags$li("Analysing the shifts in communication patterns across the two weeks.  
  
"),  
 tags$li("Focus on specific entities to determine influence within the network."),  
 tags$li("Visualising the interactions between people and vessels to explore the interactions and relationships in the knowledge graph"),  
 tags$li("Applying community detection to uncover groups that are closely associated with the community and the topic areas predominant for each group."),  
 tags$li("Detecting which person or vessel is using pseudonyms and unraveling them.  
  
"),  
 tags$li("Identifying common entities in the knowledge graph."),  
 tags$li("Understanding of the activities by unraveling pseudonyms and tracking their interactions."),  
 tags$li("Providing evidence through visual analytics to determine whether Nadia Conti is engaging in illicit activity."),  
 tags$li("Presenting a summary of Nadia’s actions and visual reasoning behind any suspicion.")  
 )  
 )))



## Section 3: Project Timeline

#-------UI-------  
 # Timeline Section  
 tags$section(id="timeline", fluidRow(column(12,  
 h2("Project Timeline"),  
 plotOutput("timeline\_plot", height="400px")  
 )))  
  
#-----server----  
  
server <- function(input, output, session) {  
   
 # --- Project Timeline ---  
 output$timeline\_plot <- renderPlot({  
 data <- read.csv(text = "  
event,group,start,end,color  
,Project Proposal,2025-06-02,2025-06-08,#a5d6a7  
,Exploratory data analysis,2025-05-20,2025-06-08,#a5d6a7  
,Exploratory data analysis,2025-06-08,2025-06-09,#DD4B39  
,R Quarto/ Netlify,2025-05-20,2025-06-08,#a5d6a7  
,R Quarto/ Netlify,2025-06-08,2025-06-30,#DD4B39  
,R Shiny App,2025-06-15,2025-07-06,#DD4B39  
,Poster,2025-06-23,2025-06-26,#DD4B39  
,User Guide,2025-06-15,2025-06-26,#DD4B39  
", stringsAsFactors = FALSE)  
 data <- data %>%  
 mutate(  
 start = as.POSIXct(start),  
 end = as.POSIXct(end),  
 color = recode(color,  
 `#a5d6a7` = "#8979BF",  
 `#DD4B39` = "#B0BF79")  
 )  
 gg\_vistime(data) +  
 geom\_vline(xintercept = as.numeric(as.POSIXct("2025-06-08")), color = "red") +  
 scale\_x\_datetime(date\_breaks = "3 days", date\_labels = "%d %b") +  
 theme\_minimal(base\_family = "Rubik") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
 }, bg = "transparent", res = 96)  
 }



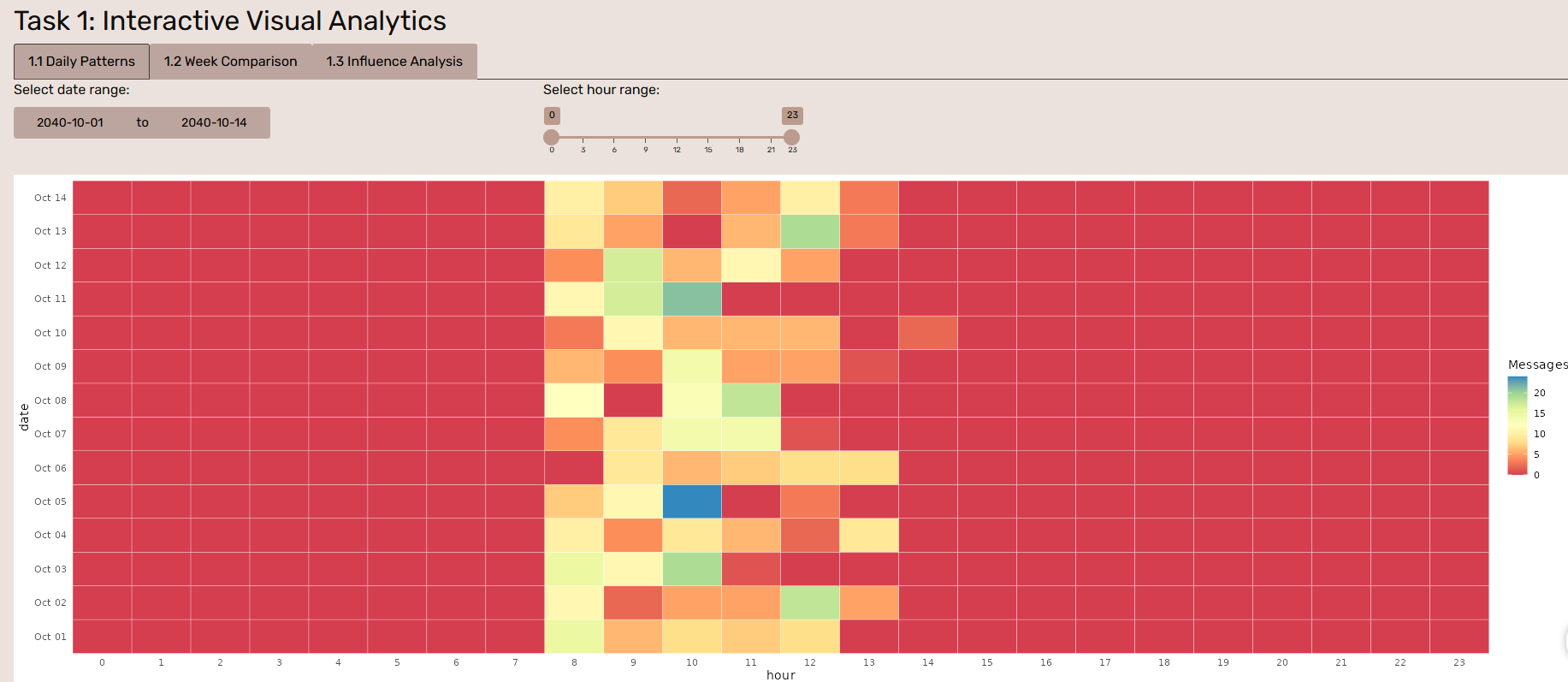
## Section 4: UI Sketch

# UI Sketch  
 tags$section(id="sketch", fluidRow(column(12,  
 h2("UI Sketch"),  
 img(src="sketch.jpg", alt="UI Sketch",  
 style="max-width:60%; border:1px solid #ccc;")



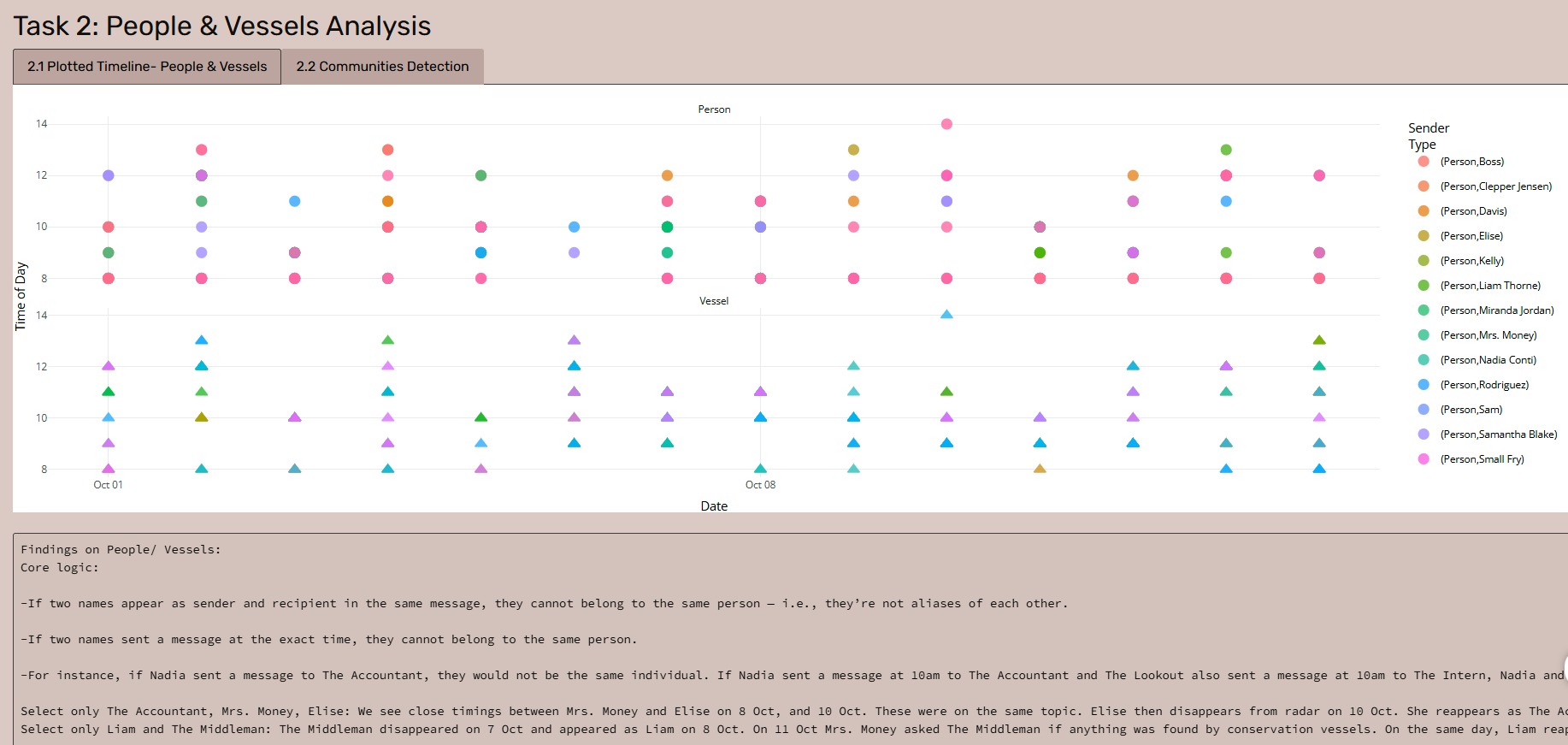
## Section 5: Task 1 Interactive Visual Analytics

#-----UI-------------  
  
 # — Task 1 —  
 tags$section(id="task1", fluidRow(column(12,  
 h2("Task 1: Interactive Visual Analytics"),  
 tabsetPanel(id="task1\_tabs", type="tabs",  
   
 tabPanel("1.1 Daily Patterns",  
 fluidRow(  
 column(4, dateRangeInput("date\_range","Select date range:", start=NULL, end=NULL)),  
 column(4, sliderInput("hour\_range","Select hour range:",0,23,c(0,23),step=1))  
 ),  
 plotOutput("heatmap", height="600px")  
 ),  
   
 tabPanel("1.2 Week Comparison",  
 fluidRow(  
 column(4, checkboxGroupInput("weeks","Select Week:",c("Week 1","Week 2"),c("Week 1","Week 2"))),  
 column(4, sliderInput("hour\_range2","Hourly interval:",0,23,c(0,23),step=1))  
 ),  
 plotOutput("week\_plot", height="450px")  
 ),  
   
 tabPanel("1.3 Influence Analysis",  
 fluidRow(  
 column(6,  
 selectInput("subtype","Select subtype:",choices=NULL),  
 plotlyOutput("inf\_plot", height="600px")  
 ),  
 column(6,  
 selectInput("target","Select entity:",choices=NULL),  
 plotOutput("net\_plot", height="600px")  
 )  
 )  
 )  
   
 )  
 )))  
   
  
  
#-----server----  
  
server <- function(input, output, session) {  
   
  
   
 # --- Load data ---  
 MC3\_graph <- fromJSON("data/MC3\_graph.json")  
 nodes\_tbl <- as\_tibble(MC3\_graph$nodes)  
 edges\_tbl <- as\_tibble(MC3\_graph$edges)  
   
 comm\_nodes <- nodes\_tbl %>%  
 filter(type=="Event", sub\_type=="Communication") %>%  
 transmute(event\_id=id, timestamp)  
 sent\_edges <- edges\_tbl %>% filter(type=="sent") %>% transmute(sender=source, event\_id=target)  
 recv\_edges <- edges\_tbl %>% filter(type=="received") %>% transmute(event\_id=source, receiver=target)  
   
 msgs <- sent\_edges %>%  
 inner\_join(recv\_edges, by="event\_id") %>%  
 inner\_join(comm\_nodes, by="event\_id") %>%  
 mutate(  
 ts = ymd\_hms(timestamp, tz="UTC"),  
 date = as\_date(ts),  
 hour = hour(ts)  
 )  
   
  
 # ---- Task 1 logic ----  
 heatmap\_data <- msgs %>%  
 count(date, hour) %>%  
 complete(date=seq(min(date),max(date),by="1 day"),  
 hour=0:23, fill=list(n=0))  
 updateDateRangeInput(session,"date\_range",  
 start=min(heatmap\_data$date),  
 end =max(heatmap\_data$date))  
   
 week\_patterns <- msgs %>%  
 mutate(week=if\_else(date<=min(date)+days(6),"Week 1","Week 2")) %>%  
 count(week,hour) %>%  
 group\_by(week) %>%  
 mutate(proportion=n/sum(n)) %>%  
 ungroup()  
   
 entity\_info <- nodes\_tbl %>%  
 filter(type=="Entity") %>%  
 transmute(id, target\_subtype=sub\_type)  
 msgs2 <- msgs %>% left\_join(entity\_info, by=c("receiver"="id"))  
 top\_inf <- msgs2 %>%  
 filter(!is.na(receiver)) %>%  
 count(target\_subtype, receiver, sender, sort=TRUE) %>%  
 group\_by(target\_subtype, receiver) %>%  
 slice\_max(n, n=10) %>%  
 ungroup()  
 subtypes <- unique(top\_inf$target\_subtype)  
 updateSelectInput(session,"subtype",  
 choices=subtypes,  
 selected=subtypes[1])  
   
 ee\_edges <- msgs %>% transmute(from=sender, to=receiver)  
 entity\_nodes <- nodes\_tbl %>% filter(type=="Entity") %>% transmute(id,name)  
 g\_ig <- graph\_from\_data\_frame(ee\_edges, vertices=entity\_nodes, directed=TRUE)  
 all\_targets <- sort(V(g\_ig)$name)  
 updateSelectInput(session,"target",  
 choices=all\_targets,  
 selected=all\_targets[1])  
   
#---- 1.1 Heat map-------------  
 filtered\_data <- reactive({  
 req(input$date\_range)  
 heatmap\_data %>%  
 filter(  
 date >= input$date\_range[1],  
 date <= input$date\_range[2],  
 hour >= input$hour\_range[1],  
 hour <= input$hour\_range[2]  
 )  
 })  
 output$heatmap <- renderPlot({  
 df <- filtered\_data()  
 ggplot(df, aes(hour, date, fill = n)) +  
 geom\_tile(color = "white") +  
 scale\_x\_continuous(breaks = seq(input$hour\_range[1], input$hour\_range[2], by = 1),  
 expand = c(0,0)) +  
 scale\_y\_date(  
 breaks = seq(min(heatmap\_data$date), max(heatmap\_data$date), by = "1 day"),  
 date\_labels = "%b %d",  
 expand = c(0, 0)  
 ) +  
 scale\_fill\_distiller(name = "Messages", palette = "Spectral", direction = 1) +  
 theme\_minimal() +  
 theme(panel.grid = element\_blank())  
 }, res = 96)  
   
   
#---------- 1.2 Week Comparison-------------  
 output$week\_plot <- renderPlot({  
 df <- week\_patterns %>%  
 filter(week %in% input$weeks,  
 hour >= input$hour\_range2[1],  
 hour <= input$hour\_range2[2])  
 ggplot(df, aes(hour, proportion, color=week)) +  
 geom\_line() + geom\_point() +  
 scale\_y\_continuous(labels=percent\_format()) +  
 theme\_light() +  
 theme(legend.position="top")  
 })  
   
 #--------------1.3 Influence Bar--------------  
 output$inf\_plot <- renderPlotly({  
 df <- top\_inf %>%  
 filter(target\_subtype == input$subtype)  
 p <- ggplot(df,  
 aes(reorder(sender, n), n,  
 fill = receiver,  
 text = paste0("Sender: ", sender, "\nCount: ", n))) +  
 geom\_col() + coord\_flip() +  
 labs(x="Sender", y="Count") +  
 theme\_minimal()  
 ggplotly(p, tooltip="text")  
 })  
   
 # 1.3 Influence Network  
 output$net\_plot <- renderPlot({  
 req(input$target)  
 root <- which(V(g\_ig)$name == input$target)  
 vids <- unique(c(  
 root,  
 neighbors(g\_ig, root, mode="in"),  
 neighbors(g\_ig, root, mode="out")  
 ))  
 subg <- induced\_subgraph(g\_ig, vids)  
 ggraph(subg, layout="kk") +  
 geom\_edge\_link(color="grey70") +  
 geom\_node\_point(aes(filter=(name==input$target)),  
 color="firebrick", size=6) +  
 geom\_node\_point(aes(filter=(name!=input$target)),  
 color="steelblue", size=4) +  
 geom\_node\_text(aes(label=name),  
 repel=TRUE, size=3) +  
 theme\_void()  
 })



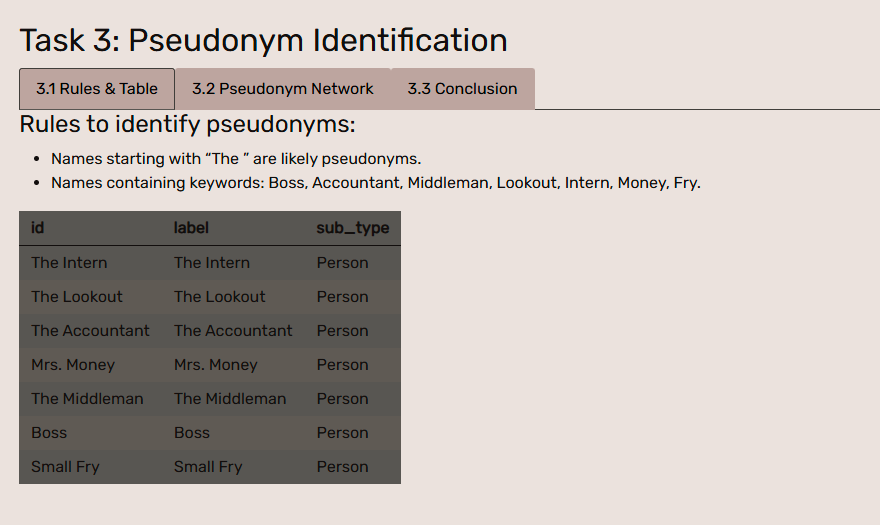
## Section 6:Task 2

#--------UI---------  
 # — Task 2 —  
 tags$section(id="task2", fluidRow(column(12,  
 h2("Task 2: People & Vessels Analysis"),  
 tabsetPanel(id="task2\_tabs", type="tabs",  
   
 tabPanel("2.1 Plotted Timeline- People & Vessels",  
 plotlyOutput("pv\_timeline", height="500px"),  
 br(), verbatimTextOutput("task2a\_findings")  
 ),  
   
 # inside your tabsetPanel(id="task2\_tabs", type="tabs", …)  
 tabPanel("2.2 Communities Detection",  
   
 # 1) Community Detection (static)  
 fluidRow(  
 column(12, h4(" Community Detection – People & Vessels")),  
 column(12, plotOutput("comm\_graph", height = "450px"))  
 ),  
   
 # 2) PageRank PageRank bar + network  
 fluidRow(  
 column(6,  
 h4("PageRank Centrality (Top 10 Nodes)"),  
 plotOutput("pr\_bar", height = "350px")  
 ),  
 column(6,  
 h4("PageRank‐Sized Network"),  
 plotOutput("pr\_graph", height = "350px")  
 )  
 ),  
 h5(" Findings"),  
 verbatimTextOutput("task2b\_pr\_findings"),  
   
 # 4) Membership summary (white background)  
 fluidRow(  
 column(12, h4("Community Group Membership – Top 5")),  
 column(12,  
 div(style="background:white; padding:8px;",  
 tableOutput("comm\_membership")  
 )  
 )  
 ),  
   
 # 5) Interactive community timeline  
 fluidRow(  
 column(12, h4("Plotted Community Timeline")),  
 column(12, plotlyOutput("comm\_timeline", height = "450px"))  
 ),  
   
 # 6) Discussion  
 h5("Discussion / Interpretation"),  
 verbatimTextOutput("task2b\_discussion")  
 )  
   
   
 )  
 )))  
  
  
  
#--------Server---------  
 # ---- Task 2 prep ----  
 g\_undir <- as\_undirected(g\_ig, mode="collapse")  
   
 comm\_obj <- reactive({ cluster\_walktrap(g\_undir) })  
   
 # Shared timeline reactive  
 entity\_meta <- nodes\_tbl %>%  
 filter(type=="Entity") %>%  
 transmute(id,name,sub\_type)  
 timeline\_df <- reactive({  
 msgs %>%  
 left\_join(entity\_meta, by=c("sender"="id")) %>%  
 filter(sub\_type %in% c("Person","Vessel")) %>%  
 mutate(Type=sub\_type)  
 })  
   
 # 2.a Interactive Timeline  
 output$pv\_timeline <- renderPlotly({  
 df <- timeline\_df()  
 p <- ggplot(df,  
 aes(date, hour,  
 color = name,  
 shape = Type,  
 text = paste(name, Type, hour))) +  
 geom\_point(size=3, alpha=0.8) +  
 facet\_wrap(~Type, ncol=1) +  
 scale\_shape\_manual(values=c(Person=16, Vessel=17)) +  
 labs(x="Date", y="Time of Day", color="Sender") +  
 theme\_minimal()  
 ggplotly(p, tooltip="text")  
 })  
 output$task2a\_findings <- renderText({  
 "Findings on People/ Vessels:  
Core logic:  
  
-If two names appear as sender and recipient in the same message, they cannot belong to the same person — i.e., they’re not aliases of each other.  
  
-If two names sent a message at the exact time, they cannot belong to the same person.  
  
-For instance, if Nadia sent a message to The Accountant, they would not be the same individual. If Nadia sent a message at 10am to The Accountant and The Lookout also sent a message at 10am to The Intern, Nadia and The Lookout cannot be the same person.  
  
Select only The Accountant, Mrs. Money, Elise: We see close timings between Mrs. Money and Elise on 8 Oct, and 10 Oct. These were on the same topic. Elise then disappears from radar on 10 Oct. She reappears as The Accountant and Mrs. Money on 11 Oct on the same topic and remains only as The Accountant till 14 Oct.  
Select only Liam and The Middleman: The Middleman disappeared on 7 Oct and appeared as Liam on 8 Oct. On 11 Oct Mrs. Money asked The Middleman if anything was found by conservation vessels. On the same day, Liam reappeared and replied Elise that nothing was found by them.  
  
Select only The Boss and Nadia: The Boss disappeared on 5 Oct and reappeared as Nadia on 8 Oct. Likely the same person.  
  
Select only Small Fry and Rodriguez: on 2 Oct Rodriguez corresponded with Remora and Mako on meeting at the slip #14. It happened again on 14 Oct as he took on dual roles and responded to the same message with different names. Likely the same person.  
  
Select only The Lookout and Sam: on 7 Oct Sam asked Kelly to get information on who authorized the permit. 2 minutes later, The Lookout (Kelly) responded to The Intern (Sam), that it was signed by Jensen from City Council.  
  
Seawatch only appeared on 10 Oct but Horizon talked to Seawatch on 8 Oct. Therefore, some other entity is Seawatch before or during 8 Oct. Defender told Seawatch on 3 Oct at 8.39am that it increased its patrol and informed Seawatch to maintain vigilance. The Lookout (Seawatch) responded to Sentinel (Defender) at 8.41am that it acknowledged the need for vigilance.  
  
 "  
 })  
   
 # 2.b Static Community Detection  
 output$comm\_net\_static <- renderPlot({  
 mem <- membership(comm\_obj())  
 ggraph(g\_undir, layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(color=factor(mem[name])), size=4) +  
 geom\_node\_text(aes(label=name, color=factor(mem[name])),  
 repel=TRUE, size=3) +  
 scale\_color\_manual(values=c("1"="#E41A1C","2"="#377EB8",  
 "3"="#4DAF4A","4"="#984EA3","5"="#FF7F00")) +  
 labs(color="Community") +  
 theme\_void()  
 })  
   
 # --- Before these outputs, define a shared reactive timeline\_df() ---  
 timeline\_df <- reactive({  
 msgs %>%  
 left\_join(  
 nodes\_tbl %>% filter(type=="Entity") %>% transmute(id,name,sub\_type),  
 by = c("sender" = "id")  
 ) %>%  
 filter(sub\_type %in% c("Person","Vessel")) %>%  
 mutate(Type = sub\_type)  
 })  
   
 # 1) Static community‐detection plot  
 output$comm\_graph <- renderPlot({  
 com <- cluster\_louvain(as.undirected(g\_ig, mode="collapse"))  
 mem <- membership(com)  
 ggraph(as.undirected(g\_ig,mode="collapse"), layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(color=factor(mem[name])), size=4) +  
 geom\_node\_text(aes(label=name, color=factor(mem[name])),  
 repel=TRUE, size=3) +  
 scale\_color\_manual(values=c("1"="#E41A1C","2"="#377EB8","3"="#4DAF4A",  
 "4"="#984EA3","5"="#FF7F00")) +  
 theme\_void()  
 })  
   
 # precompute PageRank  
 pr\_vals <- page\_rank(as.undirected(g\_ig,mode="collapse"))$vector  
   
 # 2a) Bar chart of top PageRank nodes  
 output$pr\_bar <- renderPlot({  
 pr\_df <- tibble(  
 name = V(g\_ig)$name,  
 pagerank = pr\_vals  
 ) %>%  
 arrange(desc(pagerank)) %>%  
 slice\_head(n=10)  
   
 ggplot(pr\_df, aes(reorder(name, pagerank), pagerank)) +  
 geom\_col(fill="#377EB8") +  
 coord\_flip() +  
 labs(x="Node", y="PageRank", title="Top 10 by PageRank") +  
 theme\_minimal(base\_size=12)  
 })  
   
 # 2b) Static network sized by PageRank  
 output$pr\_graph <- renderPlot({  
 mem <- membership(cluster\_louvain(as.undirected(g\_ig,mode="collapse")))  
 ggraph(as.undirected(g\_ig,mode="collapse"), layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(size=pr\_vals, color=factor(mem[name])), show.legend="point") +  
 geom\_node\_text(aes(label=name, color=factor(mem[name])),  
 repel=TRUE, size=3) +  
 scale\_size(range=c(2,8), name="PageRank") +  
 scale\_color\_manual(values=c("1"="#E41A1C","2"="#377EB8","3"="#4DAF4A",  
 "4"="#984EA3","5"="#FF7F00")) +  
 theme\_void()  
 })  
 # 3) Findings for PageRank step  
 output$task2b\_pr\_findings <- renderText({  
 "Findings:  
- There were 5 closely associated groups. Community 5 (Clepper and Miranda) appeared to be segmented from the central group, due to the non-involvement from the nature of their investigative work.  
From the graph, we extracted the 8 influential nodes to focus on:  
Community 1: Mako  
Community 2: Neptune, Remora, Nadia, Davis  
Community 3: N/A as they were not very influential at global level  
Community 4: Mrs. Money, Boss, The Middleman  
Community 5: N/A as they were not very influential at global level  
Community 6: N/A as they were not very influential at global level"  
 })  
   
 # 4) Membership & top topics (top 5 groups)  
 output$comm\_membership <- renderTable({  
 com <- cluster\_louvain(as.undirected(g\_ig,mode="collapse"))  
 mem <- membership(com)  
 # build membership + topics as before  
 mem\_df <- tibble(name=names(mem), community=mem)  
 part <- edges\_tbl %>%  
 filter(type %in% c("sent","received")) %>%  
 mutate(entity = ifelse(type=="sent", source, target),  
 event = ifelse(type=="sent", target, source)) %>%  
 select(entity, event)  
 about <- edges\_tbl %>% filter(type=="about") %>% transmute(event=source, topic=target)  
 topics <- nodes\_tbl %>% filter(type=="Topic") %>% transmute(id, topic\_name=name)  
   
 joined <- part %>%  
 inner\_join(about, by="event") %>%  
 inner\_join(topics, by=c("topic"="id")) %>%  
 left\_join(mem\_df, by=c("entity"="name"))  
   
 top\_topics <- joined %>%  
 count(community, topic\_name, name="n") %>%  
 group\_by(community) %>%  
 slice\_max(n, n=3) %>%  
 summarise(predominant = paste(topic\_name, collapse=", "))  
   
 members <- mem\_df %>%  
 group\_by(community) %>%  
 summarise(Members=paste(name, collapse=", "))  
   
 out <- members %>%  
 left\_join(top\_topics, by="community") %>%  
 rename(Group=community) %>%  
 arrange(Group) %>%  
 slice\_head(n=5)  
   
 out  
 }, bordered=TRUE, spacing="xs")  
   
  
   
 # 6) Interactive community timeline  
 output$comm\_timeline <- renderPlotly({  
 # reuse timeline\_df() defined above Task 2.a  
 df <- timeline\_df() %>%  
 left\_join(tibble(name=names(membership(cluster\_louvain(as.undirected(g\_ig,mode="collapse")))),  
 community=membership(cluster\_louvain(as.undirected(g\_ig,mode="collapse")))),  
 by="name")  
 p <- ggplot(df, aes(date, hour, color=factor(community), shape=Type,  
 text=paste(name, "Grp", community, hour))) +  
 geom\_point(size=3, alpha=0.8) +  
 facet\_wrap(~Type, ncol=1) +  
 scale\_shape\_manual(values=c(Person=16, Vessel=17)) +  
 labs(x="Date", y="Time of Day", color="Community") +  
 theme\_minimal()  
 ggplotly(p, tooltip="text")  
 })  
   
 # 7) Discussion / Interpretation  
 output$task2b\_discussion <- renderText({  
 "Discussion:  
We mainly focused on the conversations by 8 influential nodes and some related nodes:  
Conservation Group (Comm.1): Samantha Blake informed Mako to stop operations on 8 and 10th Oct. Serenity is a private luxury yacht. Osprey was likely a tourism vessel looking for charter from Mako for their tourists.  
  
Permit (Comm. 2): Neptune, Remora, Nadia, and Davis were working on Nemo Reef operation. This referred to the Music Video Production for Sailor Shift on 14 Oct.  
  
Pseudonym (Comm. 3): Other than communicating among themselves, The Lookout appeared to have also externally corresponded with Sentinel, Reef Guardian and Horizon (conservation based topics), while The Intern also externally corresponded with Mrs. Money.  
  
Suspicious (Comm. 4): The Middleman had access to Council documents. Mrs. Money had funding from sources that would not flag out to regulators for her operations. Mrs. Money was investigating V. Miesel’s structures. On 5 Oct, Boss told Mrs. Money to disguise financial trails through tourism ventures and destroy evidence of Nemo Reef operations.  
  
Hacklee Herald (Comm. 5): Conversations between Clepper and his intern Miranda which ended on 11 Oct. Miranda mentioned an Oceanus City Council Member meeting with unmarked vessels at night."  
 })



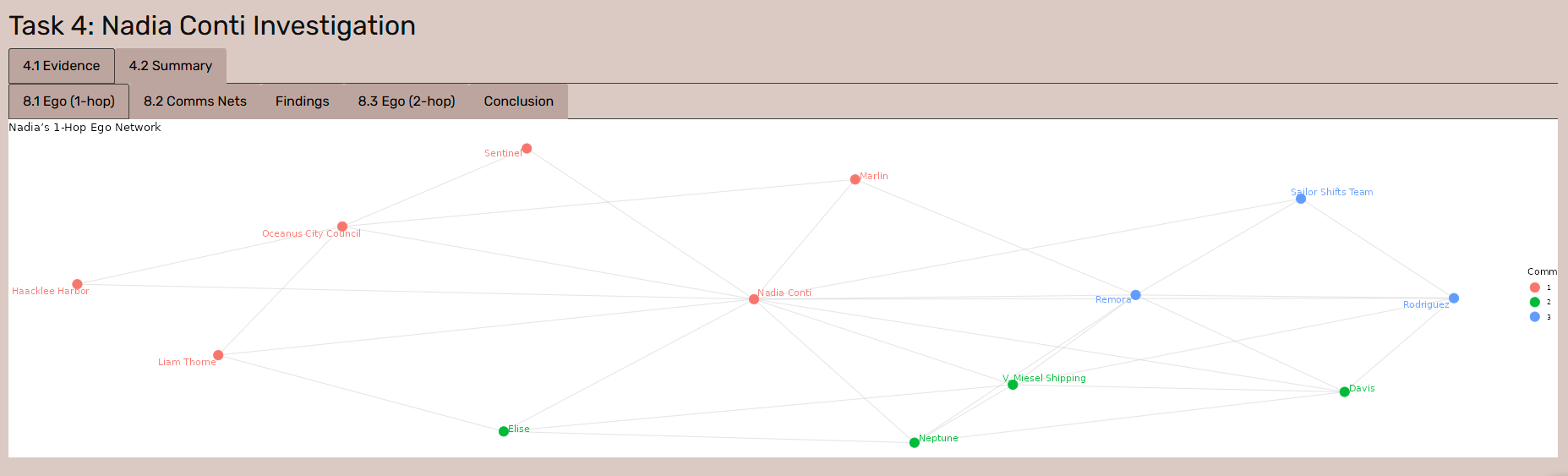
## Section 7:Task 3 Pseudonym Identification

#-------------UI------------------  
 # — Task 3: Pseudonym Identification —  
 tags$section(id="task3", fluidRow(  
 column(12,  
 h2("Task 3: Pseudonym Identification"),  
 tabsetPanel(id="task3\_tabs", type="tabs",  
   
 # 3.1: Rules & Table  
 tabPanel("3.1 Rules & Table",  
 h4("Rules to identify pseudonyms:"),  
 tags$ul(  
 tags$li("Names starting with “The ” are likely pseudonyms."),  
 tags$li("Names containing keywords: Boss, Accountant, Middleman, Lookout, Intern, Money, Fry.")  
 ),  
 tableOutput("pseudo\_table")  
 ),  
   
 # 3.2: Pseudonym Network  
 tabPanel("3.2 Pseudonym Network",  
 visNetworkOutput("pseudonym\_net", height = "500px")  
 ),  
   
 # 3.3: Conclusion  
 tabPanel("3.3 Conclusion",  
 verbatimTextOutput("task3\_conclusion")  
 )  
   
 )  
 )  
 ))  
  
  
#-----server--------  
   
 # 3.1 Build the pseudo table  
 nodes\_person <- nodes\_tbl %>%  
 filter(sub\_type == "Person") %>%  
 transmute(id, label = name, sub\_type)  
   
 pseudonym\_keywords <- c("Boss", "Accountant", "Middleman", "Lookout",  
 "Intern", "Money", "Fry")  
   
 pseudo\_df <- nodes\_person %>%  
 mutate(  
 is\_prefix = str\_detect(label, "^The "),  
 is\_keyword = str\_detect(label, paste(pseudonym\_keywords, collapse = "|")),  
 is\_pseudonym = is\_prefix | is\_keyword  
 ) %>%  
 filter(is\_pseudonym) %>%  
 select(id, label, sub\_type)  
   
 output$pseudo\_table <- renderTable({  
 pseudo\_df  
 }, striped = TRUE, bordered = TRUE)  
   
 # 3.2 Build & render the pseudonym-only communication network  
 pseudo\_labels <- pseudo\_df$label  
   
 pv\_nodes <- data.frame(id = pseudo\_labels, label = pseudo\_labels)  
   
 pv\_edges <- msgs %>%  
 # attach sender label  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("sender" = "id")) %>%  
 rename(from = label) %>%  
 # attach receiver label  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("receiver" = "id")) %>%  
 rename(to = label) %>%  
 filter(from %in% pseudo\_labels, to %in% pseudo\_labels) %>%  
 distinct(from, to)  
   
 output$pseudonym\_net <- renderVisNetwork({  
 visNetwork(pv\_nodes, pv\_edges) %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)  
 })  
   
 # 3.3 Conclusion  
 output$task3\_conclusion <- renderText({  
 paste(  
 "→ Identified pseudonyms:",  
 paste(pseudo\_df$label, collapse = ", "),  
 "",  
 "With the explicit identification of pseudonyms such as “Boss,” “The Intern,” “The Accountant,” “Mrs. Money,” “The Middleman,” “The Lookout,” and “Small Fry,” my understanding of the activities within the Oceanus network changed significantly.",  
 "Initially, communications and relationships appeared to be fragmented among many individuals and vessels. However, after mapping pseudonyms onto the network, several important patterns emerged:",  
 "",  
 "-Pseudonyms as Key Hubs: In the visual network, the detected pseudonyms are not peripheral actors—they consistently occupy central hub positions, connecting to multiple vessels and people through various types of relationships (such as “Reports,” “Coordinates,” “Operates,” and “Suspicious”).",  
 "-Obscured True Roles: What previously looked like many unrelated participants was actually a small set of influential actors operating under multiple pseudonyms. For example, “The Intern” and “Boss” are both connected to several vessels via “Coordinates” and “Operates” relationships, indicating they may be orchestrating or managing operations behind the scenes.).",  
 "-Concealment of Illicit Activity: The structure revealed that pseudonyms are often used to mask direct links between real individuals and vessels—particularly in suspicious or high-frequency communication events. This suggests intentional obfuscation, possibly to hide illicit coordination or the flow of sensitive information.)",  
 "-Redefining Influence and Risk: By highlighting pseudonyms, it becomes clear that influence within the network is far more centralized than the real-name view suggested. A few pseudonyms control or interact with a large proportion of vessels and people, making them high-value targets for further investigation.",  
 sep = "\n")  
 })



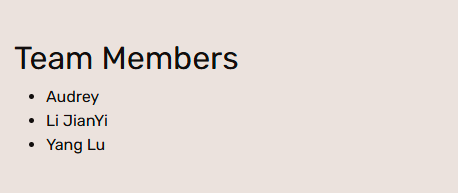
## Section 8:Task 4 Nidia Conti Investigation

#------------UI-------------  
 tags$section(id="task4", fluidRow(  
 column(12,  
 h2("Task 4: Nadia Conti Investigation"),  
 tabsetPanel(id="task4\_tabs", type="tabs",  
   
 # 4.1 Evidence (5 steps) —  
 tabPanel("4.1 Evidence",  
 tabsetPanel(type="tabs",  
   
 # 1) 8.1 Nadia’s Ego Network (1-hop Louvain)  
 tabPanel("8.1 Ego (1-hop)",  
 plotOutput("nadia\_ego\_1hop", height="400px")  
 ),  
   
 # 2) 8.2 Sent vs Received Comms Nets  
 tabPanel("8.2 Comms Nets",  
 fluidRow(  
 column(6,  
 h4("8.2.2.1 Sent Communications Network"),  
 visNetworkOutput("nadia\_sent\_net", height="350px")  
 ),  
 column(6,  
 h4("8.2.2.2 Received Communications Network"),  
 visNetworkOutput("nadia\_recv\_net", height="350px")  
 )  
 )  
 ),  
   
 # 3) Findings  
 tabPanel("Findings",  
 verbatimTextOutput("task4\_1\_findings")  
 ),  
   
 # 4) 8.3 Nadia’s 2-Hop Ego (Degree Centrality)  
 tabPanel("8.3 Ego (2-hop)",  
 plotOutput("nadia\_ego\_2hop", height="400px")  
 ),  
   
 # 5) Conclusion  
 tabPanel("Conclusion",  
 verbatimTextOutput("task4\_1\_conclusion")  
 )  
   
 )  
 ),  
   
 # 4.2 Summary (2 steps) —  
 tabPanel("4.2 Summary",  
 fluidRow(  
 column(6,  
 h4("8.4.2 Plotted Timeline (2-hop)"),  
 plotlyOutput("nadia\_2hop\_timeline", height="350px")  
 ),  
 column(6,  
 h4("8.5.1 Ego & Comms Table"),  
 DT::dataTableOutput("nadia\_comm\_table")  
 )  
 )  
 )  
   
 )  
 )  
 ))  
  
#------------Server-------------  
 # 4.0 Nadia’s node ID  
 nadia\_id <- nodes\_tbl %>% filter(name == "Nadia Conti") %>% pull(id)  
   
 # — 4.1.1 Nadia’s 1-Hop Ego Network (Louvain) —  
 ego1 <- make\_ego\_graph(as.undirected(g\_ig, mode="collapse"),  
 order = 1,  
 nodes = which(V(g\_ig)$name == "Nadia Conti"))[[1]]  
 com1 <- cluster\_louvain(ego1)  
 mem1 <- membership(com1)  
   
 output$nadia\_ego\_1hop <- renderPlot({  
 ggraph(ego1, layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(color = factor(mem1)), size=5) +  
 geom\_node\_text(aes(label = name, color = factor(mem1)),  
 repel=TRUE) +  
 labs(title = "Nadia’s 1-Hop Ego Network", color = "Comm") +  
 theme\_void()  
 })  
   
 # — 4.1.2 Sent vs Received Communications Networks —  
   
 # prepare sent edges/nodes  
 sent\_edges1 <- msgs %>%  
 filter(sender == nadia\_id) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("sender" = "id")) %>% rename(from = label) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("receiver" = "id")) %>% rename(to = label) %>%  
 distinct(from, to)  
 sent\_nodes1 <- tibble(id = unique(c(sent\_edges1$from, sent\_edges1$to)),  
 label = id)  
   
 output$nadia\_sent\_net <- renderVisNetwork({  
 g\_sent <- graph\_from\_data\_frame(sent\_edges1,  
 vertices = sent\_nodes1,  
 directed = TRUE)  
 visIgraph(g\_sent, layout = "layout\_with\_fr") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)  
 })  
   
 # prepare received edges/nodes  
 recv\_edges1 <- msgs %>%  
 filter(receiver == nadia\_id) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("sender" = "id")) %>% rename(from = label) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("receiver" = "id")) %>% rename(to = label) %>%  
 distinct(from, to)  
 recv\_nodes1 <- tibble(id = unique(c(recv\_edges1$from, recv\_edges1$to)),  
 label = id)  
   
 output$nadia\_recv\_net <- renderVisNetwork({  
 g\_recv <- graph\_from\_data\_frame(recv\_edges1,  
 vertices = recv\_nodes1,  
 directed = TRUE)  
 visIgraph(g\_recv, layout = "layout\_with\_fr") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)  
 })  
   
 # — 4.1.3 Findings —  
 output$task4\_1\_findings <- renderText({  
 paste0(  
 "Nodes involved: Nadia, Davis, Elise, Haacklee Harbor, Liam, Marlin,\n",  
 "Neptune, Oceanus City Council, Remora, Rodriguez,\n",  
 "Sailor Shifts Team, Sentinel, V. Miesel Shipping.\n\n",  
 "• Sent more direct communications to Liam (2) and Neptune (2).\n",  
 "• Received more communications from Davis (5), Elise (3), Liam (2).\n\n",  
 "Next step: drill into suspicious chars/activities."  
 )  
 })  
   
 # — 4.1.4 Nadia’s 2-Hop Ego (Degree Centrality) —  
 ego2 <- make\_ego\_graph(as.undirected(g\_ig, mode="collapse"),  
 order = 2,  
 nodes = which(V(g\_ig)$name == "Nadia Conti"))[[1]]  
 deg2 <- degree(ego2)  
 deg2\_df <- tibble(name = names(deg2), degree = deg2)  
   
 output$nadia\_ego\_2hop <- renderPlot({  
 ggplot(deg2\_df, aes(reorder(name, degree), degree)) +  
 geom\_col() +  
 coord\_flip() +  
 labs(title = "Nadia’s 2-Hop Ego (Degree Centrality)",  
 x = "Node", y = "Degree") +  
 theme\_minimal()  
 })  
   
 # — 4.1.5 Conclusion —  
 output$task4\_1\_conclusion <- renderText({  
 paste0(  
 "1-Hop nodes: Nadia, Davis, Elise, Haacklee Harbor, Liam, Marlin,\n",  
 "Neptune, Oceanus City Council, Remora, Rodriguez,\n",  
 "Sailor Shifts Team, Sentinel, V. Miesel Shipping.\n\n",  
 "2-Hop additions: EcoVigil, Sam, The Accountant, Nemo Reef.\n\n",  
 "Use these 2-hop nodes to drill deeper for suspicious links."  
 )  
 })  
   
 # — 4.2.1 Plotted Timeline for all 2-Hop actors —  
 hop2\_names <- V(ego2)$name  
 id\_map <- nodes\_tbl %>% filter(name %in% hop2\_names) %>% transmute(id, name)  
 timeline2 <- msgs %>%  
 filter(sender %in% id\_map$id | receiver %in% id\_map$id) %>%  
 mutate(Actor = if\_else(sender %in% id\_map$id,  
 id\_map$name[match(sender, id\_map$id)],  
 id\_map$name[match(receiver, id\_map$id)]))  
   
 output$nadia\_2hop\_timeline <- renderPlotly({  
 p <- ggplot(timeline2, aes(date, hour, color = Actor,  
 text = paste0(Actor, "\n", format(ts, "%Y-%m-%d %H:%M")))) +  
 geom\_point(size = 3, alpha = 0.8) +  
 labs(x = "Date", y = "Hour", color = "Actor") +  
 theme\_minimal()  
 ggplotly(p, tooltip = "text")  
 })  
   
 # — 4.2.2 Ego & Comms Table for 2-Hop actors —  
 comm2\_table <- timeline2 %>%  
 transmute(  
 Date = as.character(date),  
 Time = format(ts, "%H:%M"),  
 Sender = nodes\_tbl$name[match(sender, nodes\_tbl$id)],  
 Receiver = nodes\_tbl$name[match(receiver, nodes\_tbl$id)],  
 EventID = event\_id  
 )  
   
 output$nadia\_comm\_table <- DT::renderDataTable({  
 DT::datatable(comm2\_table,  
 options = list(pageLength = 10, autoWidth = TRUE),  
 rownames = FALSE)  
 })



## Section 9: Team introduction

#---------UI---------  
 tags$section(id="team", fluidRow(column(12,  
 h2("Team Members"),  
 tags$ul(  
 tags$li("Audrey"),  
 tags$li("Li JianYi"),  
 tags$li("Yang Lu")  
 )  
 )))  
)



## Section 10: The Entire Shiny Code Chunk

The final UI link: https://roxyyanglu.shinyapps.io/Take-home\_Ex03/

library(shiny)  
library(visNetwork)  
library(ggraph)  
library(plotly)  
library(jsonlite)  
library(tidyverse)  
library(lubridate)  
library(igraph)  
library(vistime)  
library(bslib)  
library(scales)  
library(DT)  
library(stringr)  
library(ggplot2)  
  
# general HTTP downloads  
options(timeout = max(600, getOption("timeout")))  
# rsconnect-specific network operations  
options(rsconnect.http.timeout = 120)  
  
  
# --------------------------------------------------  
# 1. DIY bslib theme  
# --------------------------------------------------  
theme\_covert <- bs\_theme(  
 bg = "#585651",  
 fg = "#090605",  
 primary = "#ba9b8d",  
 secondary = "#814a38",  
 base\_font = font\_google("Rubik"),  
 heading\_font = font\_google("Rubik")  
)  
  
# --------------------------------------------------  
# UI  
# --------------------------------------------------  
ui <- fluidPage(  
 theme = theme\_covert,  
 tags$head(  
 tags$style(HTML("  
 html, body { scroll-behavior: smooth; }  
 .navbar-covert { background-color: #FFF; }  
 .navbar-covert .nav-link { color: #c17e72 !important; font-weight: 500; }  
 .navbar-covert .nav-link:hover { color: #DBCAC3 !important; }  
 body { padding-top: 70px; }  
 section { padding: 60px 20px; }  
 /\* web BG set for #DBCAC3 section \*/  
#home,  
#sketch,  
#task2,  
#task4 {  
 background-color: #DBCAC3 !important;  
}  
  
/\* web BG set for #ebe2de section \*/  
#timeline,  
#task1,  
#task3,  
#team {  
 background-color: #ebe2de !important;  
}  
  
 /\* TabsetPanel \*/  
 .nav-tabs .nav-link {  
 color: #000 !important;  
 background-color: #bda59f !important;  
 }  
 .nav-tabs .nav-item.show .nav-link,  
 .nav-tabs .nav-link.active {  
 color: #000 !important;  
 background-color: #bda59f !important;  
 }  
  
 /\* Dropdown box color matching \*/  
 .form-control.shiny-bound-input,  
 .selectize-control.single .selectize-input {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 border-color: #bda59f !important;  
 }  
 .selectize-dropdown {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 }  
  
 /\* dateRangeInput bg & text \*/  
 #date\_range .form-control {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 border-color: #bda59f !important;  
 }  
 #date\_range .input-group-text {  
 background-color: #bda59f !important;  
 color: #000 !important;  
 border-color: #bda59f !important;page  
 }  
 "))  
 ),  
   
 # Navbar  
 tags$nav(class = "navbar navbar-expand-lg navbar-covert fixed-top",  
 tags$div(class = "container-fluid",  
 tags$a(class = "navbar-brand text-white", href = "#home",  
 tags$span("COVERT REEF "), tags$span("\U0001FAB8", style = "font-size:24px;")  
 ),  
 tags$button(class = "navbar-toggler", type = "button",  
 `data-bs-toggle` = "collapse", `data-bs-target` = "#menuCovert",  
 tags$span(class = "navbar-toggler-icon")  
 ),  
 tags$div(class = "collapse navbar-collapse", id = "menuCovert",  
 tags$ul(class = "navbar-nav ms-auto",  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#home", "Home")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#timeline","Timeline")),  
 tags$li(class= "nav-item", tags$a(class="nav-link", href="#sketch", "UI Sketch")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task1", "Task 1")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task2", "Task 2")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task3", "Task 3")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#task4", "Task 4")),  
 tags$li(class = "nav-item", tags$a(class = "nav-link", href = "#team", "Team"))  
 )  
 )  
 )  
 ),  
   
 # Home Section  
 tags$section(id="home", fluidRow(column(12,  
 h1("Project Introduction"),  
 p("Clepper, a lead investigator on Oceanus, has been closely monitoring the closure of Nemo Reef. Over the span of two weeks, he and his intern listened to and analyzed radio communications and utilized his investigative tools to uncover a complex web of expedited approvals, hidden communication, and secretive logistics. Their investigation revealed a story involving high-level Oceanus officials, Sailor Shift’s team, local influential families, and the Green Guardians, who are a local conservationist group, pointing towards possibilities of corruption and manipulation. Amidst this investigation, Nadia Conti, a known figure formerly entangled in illegal fishing operations, has resurfaced as a person of interest. The project aims to develop new and novel visual analytics techniques to support Clepper’s investigation in uncovering the full extent of the events on Oceanus."),  
 h3("Objectives"),  
 tags$ul(  
 tags$li("Identifying daily temporal patterns in communication to detect recurring message timings.  
  
"),  
 tags$li("Analysing the shifts in communication patterns across the two weeks.  
  
"),  
 tags$li("Focus on specific entities to determine influence within the network."),  
 tags$li("Visualising the interactions between people and vessels to explore the interactions and relationships in the knowledge graph"),  
 tags$li("Applying community detection to uncover groups that are closely associated with the community and the topic areas predominant for each group."),  
 tags$li("Detecting which person or vessel is using pseudonyms and unraveling them.  
  
"),  
 tags$li("Identifying common entities in the knowledge graph."),  
 tags$li("Understanding of the activities by unraveling pseudonyms and tracking their interactions."),  
 tags$li("Providing evidence through visual analytics to determine whether Nadia Conti is engaging in illicit activity."),  
 tags$li("Presenting a summary of Nadia’s actions and visual reasoning behind any suspicion.")  
 )  
 ))),  
   
 # Timeline Section  
 tags$section(id="timeline", fluidRow(column(12,  
 h2("Project Timeline"),  
 plotOutput("timeline\_plot", height="400px")  
 ))),  
   
 # UI Sketch  
 tags$section(id="sketch", fluidRow(column(12,  
 h2("UI Sketch"),  
 img(src="sketch.jpg", alt="UI Sketch",  
 style="max-width:60%; border:1px solid #ccc;")  
 ))),  
   
 # — Task 1 —  
 tags$section(id="task1", fluidRow(column(12,  
 h2("Task 1: Interactive Visual Analytics"),  
 tabsetPanel(id="task1\_tabs", type="tabs",  
   
 tabPanel("1.1 Daily Patterns",  
 fluidRow(  
 column(4, dateRangeInput("date\_range","Select date range:", start=NULL, end=NULL)),  
 column(4, sliderInput("hour\_range","Select hour range:",0,23,c(0,23),step=1))  
 ),  
 plotOutput("heatmap", height="600px")  
 ),  
   
 tabPanel("1.2 Week Comparison",  
 fluidRow(  
 column(4, checkboxGroupInput("weeks","Select Week:",c("Week 1","Week 2"),c("Week 1","Week 2"))),  
 column(4, sliderInput("hour\_range2","Hourly interval:",0,23,c(0,23),step=1))  
 ),  
 plotOutput("week\_plot", height="450px")  
 ),  
   
 tabPanel("1.3 Influence Analysis",  
 fluidRow(  
 column(6,  
 selectInput("subtype","Select subtype:",choices=NULL),  
 plotlyOutput("inf\_plot", height="600px")  
 ),  
 column(6,  
 selectInput("target","Select entity:",choices=NULL),  
 plotOutput("net\_plot", height="600px")  
 )  
 )  
 )  
   
 )  
 ))),  
   
 # — Task 2 —  
 tags$section(id="task2", fluidRow(column(12,  
 h2("Task 2: People & Vessels Analysis"),  
 tabsetPanel(id="task2\_tabs", type="tabs",  
   
 tabPanel("2.1 Plotted Timeline- People & Vessels",  
 plotlyOutput("pv\_timeline", height="500px"),  
 br(), verbatimTextOutput("task2a\_findings")  
 ),  
   
 # inside your tabsetPanel(id="task2\_tabs", type="tabs", …)  
 tabPanel("2.2 Communities Detection",  
   
 # 1) Community Detection (static)  
 fluidRow(  
 column(12, h4(" Community Detection – People & Vessels")),  
 column(12, plotOutput("comm\_graph", height = "450px"))  
 ),  
   
 # 2) PageRank PageRank bar + network  
 fluidRow(  
 column(6,  
 h4("PageRank Centrality (Top 10 Nodes)"),  
 plotOutput("pr\_bar", height = "350px")  
 ),  
 column(6,  
 h4("PageRank‐Sized Network"),  
 plotOutput("pr\_graph", height = "350px")  
 )  
 ),  
 h5(" Findings"),  
 verbatimTextOutput("task2b\_pr\_findings"),  
   
 # 4) Membership summary (white background)  
 fluidRow(  
 column(12, h4("Community Group Membership – Top 5")),  
 column(12,  
 div(style="background:white; padding:8px;",  
 tableOutput("comm\_membership")  
 )  
 )  
 ),  
   
 # 5) Interactive community timeline  
 fluidRow(  
 column(12, h4("Plotted Community Timeline")),  
 column(12, plotlyOutput("comm\_timeline", height = "450px"))  
 ),  
   
 # 6) Discussion  
 h5("Discussion / Interpretation"),  
 verbatimTextOutput("task2b\_discussion")  
 )  
   
   
 )  
 ))),  
   
 # — Task 3: Pseudonym Identification —  
 tags$section(id="task3", fluidRow(  
 column(12,  
 h2("Task 3: Pseudonym Identification"),  
 tabsetPanel(id="task3\_tabs", type="tabs",  
   
 # 3.1: Rules & Table  
 tabPanel("3.1 Rules & Table",  
 h4("Rules to identify pseudonyms:"),  
 tags$ul(  
 tags$li("Names starting with “The ” are likely pseudonyms."),  
 tags$li("Names containing keywords: Boss, Accountant, Middleman, Lookout, Intern, Money, Fry.")  
 ),  
 tableOutput("pseudo\_table")  
 ),  
   
 # 3.2: Pseudonym Network  
 tabPanel("3.2 Pseudonym Network",  
 visNetworkOutput("pseudonym\_net", height = "500px")  
 ),  
   
 # 3.3: Conclusion  
 tabPanel("3.3 Conclusion",  
 verbatimTextOutput("task3\_conclusion")  
 )  
   
 )  
 )  
 )),  
   
  
   
  
 ## — Task 4: Nadia Conti Investigation —  
 ## ——————— UI ————————  
 tags$section(id="task4", fluidRow(  
 column(12,  
 h2("Task 4: Nadia Conti Investigation"),  
 tabsetPanel(id="task4\_tabs", type="tabs",  
   
 # 4.1 Evidence (5 steps) —  
 tabPanel("4.1 Evidence",  
 tabsetPanel(type="tabs",  
   
 # 1) 8.1 Nadia’s Ego Network (1-hop Louvain)  
 tabPanel("8.1 Ego (1-hop)",  
 plotOutput("nadia\_ego\_1hop", height="400px")  
 ),  
   
 # 2) 8.2 Sent vs Received Comms Nets  
 tabPanel("8.2 Comms Nets",  
 fluidRow(  
 column(6,  
 h4("8.2.2.1 Sent Communications Network"),  
 visNetworkOutput("nadia\_sent\_net", height="350px")  
 ),  
 column(6,  
 h4("8.2.2.2 Received Communications Network"),  
 visNetworkOutput("nadia\_recv\_net", height="350px")  
 )  
 )  
 ),  
   
 # 3) Findings  
 tabPanel("Findings",  
 verbatimTextOutput("task4\_1\_findings")  
 ),  
   
 # 4) 8.3 Nadia’s 2-Hop Ego (Degree Centrality)  
 tabPanel("8.3 Ego (2-hop)",  
 plotOutput("nadia\_ego\_2hop", height="400px")  
 ),  
   
 # 5) Conclusion  
 tabPanel("Conclusion",  
 verbatimTextOutput("task4\_1\_conclusion")  
 )  
   
 )  
 ),  
   
 # 4.2 Summary (2 steps) —  
 tabPanel("4.2 Summary",  
 fluidRow(  
 column(6,  
 h4("8.4.2 Plotted Timeline (2-hop)"),  
 plotlyOutput("nadia\_2hop\_timeline", height="350px")  
 ),  
 column(6,  
 h4("8.5.1 Ego & Comms Table"),  
 DT::dataTableOutput("nadia\_comm\_table")  
 )  
 )  
 )  
   
 )  
 )  
 )),  
   
  
   
 #Team introduction  
 tags$section(id="team", fluidRow(column(12,  
 h2("Team Members"),  
 tags$ul(  
 tags$li("Audrey"),  
 tags$li("Li JianYi"),  
 tags$li("Yang Lu")  
 )  
 )))  
)  
  
  
  
  
  
  
  
  
  
# --------------------------------------------------  
# Server  
# --------------------------------------------------  
server <- function(input, output, session) {  
   
 # --- Task 0: Timeline ---  
 output$timeline\_plot <- renderPlot({  
 data <- read.csv(text = "  
event,group,start,end,color  
,Project Proposal,2025-06-02,2025-06-08,#a5d6a7  
,Exploratory data analysis,2025-05-20,2025-06-08,#a5d6a7  
,Exploratory data analysis,2025-06-08,2025-06-09,#DD4B39  
,R Quarto/ Netlify,2025-05-20,2025-06-08,#a5d6a7  
,R Quarto/ Netlify,2025-06-08,2025-06-30,#DD4B39  
,R Shiny App,2025-06-15,2025-07-06,#DD4B39  
,Poster,2025-06-23,2025-06-26,#DD4B39  
,User Guide,2025-06-15,2025-06-26,#DD4B39  
", stringsAsFactors = FALSE)  
 data <- data %>%  
 mutate(  
 start = as.POSIXct(start),  
 end = as.POSIXct(end),  
 color = recode(color,  
 `#a5d6a7` = "#8979BF",  
 `#DD4B39` = "#B0BF79")  
 )  
 gg\_vistime(data) +  
 geom\_vline(xintercept = as.numeric(as.POSIXct("2025-06-08")), color = "red") +  
 scale\_x\_datetime(date\_breaks = "3 days", date\_labels = "%d %b") +  
 theme\_minimal(base\_family = "Rubik") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
 }, bg = "transparent", res = 96)  
   
 # --- Load & preprocess for Task 1 & 2 ---  
 MC3\_graph <- fromJSON("data/MC3\_graph.json")  
 nodes\_tbl <- as\_tibble(MC3\_graph$nodes)  
 edges\_tbl <- as\_tibble(MC3\_graph$edges)  
   
 comm\_nodes <- nodes\_tbl %>%  
 filter(type=="Event", sub\_type=="Communication") %>%  
 transmute(event\_id=id, timestamp)  
 sent\_edges <- edges\_tbl %>% filter(type=="sent") %>% transmute(sender=source, event\_id=target)  
 recv\_edges <- edges\_tbl %>% filter(type=="received") %>% transmute(event\_id=source, receiver=target)  
   
 msgs <- sent\_edges %>%  
 inner\_join(recv\_edges, by="event\_id") %>%  
 inner\_join(comm\_nodes, by="event\_id") %>%  
 mutate(  
 ts = ymd\_hms(timestamp, tz="UTC"),  
 date = as\_date(ts),  
 hour = hour(ts)  
 )  
   
   
   
   
   
   
   
 # ---- Task 1 logic ----  
 heatmap\_data <- msgs %>%  
 count(date, hour) %>%  
 complete(date=seq(min(date),max(date),by="1 day"),  
 hour=0:23, fill=list(n=0))  
 updateDateRangeInput(session,"date\_range",  
 start=min(heatmap\_data$date),  
 end =max(heatmap\_data$date))  
   
 week\_patterns <- msgs %>%  
 mutate(week=if\_else(date<=min(date)+days(6),"Week 1","Week 2")) %>%  
 count(week,hour) %>%  
 group\_by(week) %>%  
 mutate(proportion=n/sum(n)) %>%  
 ungroup()  
   
 entity\_info <- nodes\_tbl %>%  
 filter(type=="Entity") %>%  
 transmute(id, target\_subtype=sub\_type)  
 msgs2 <- msgs %>% left\_join(entity\_info, by=c("receiver"="id"))  
 top\_inf <- msgs2 %>%  
 filter(!is.na(receiver)) %>%  
 count(target\_subtype, receiver, sender, sort=TRUE) %>%  
 group\_by(target\_subtype, receiver) %>%  
 slice\_max(n, n=10) %>%  
 ungroup()  
 subtypes <- unique(top\_inf$target\_subtype)  
 updateSelectInput(session,"subtype",  
 choices=subtypes,  
 selected=subtypes[1])  
   
 ee\_edges <- msgs %>% transmute(from=sender, to=receiver)  
 entity\_nodes <- nodes\_tbl %>% filter(type=="Entity") %>% transmute(id,name)  
 g\_ig <- graph\_from\_data\_frame(ee\_edges, vertices=entity\_nodes, directed=TRUE)  
 all\_targets <- sort(V(g\_ig)$name)  
 updateSelectInput(session,"target",  
 choices=all\_targets,  
 selected=all\_targets[1])  
   
 # 1.1 Heatmap  
 filtered\_data <- reactive({  
 req(input$date\_range)  
 heatmap\_data %>%  
 filter(  
 date >= input$date\_range[1],  
 date <= input$date\_range[2],  
 hour >= input$hour\_range[1],  
 hour <= input$hour\_range[2]  
 )  
 })  
 output$heatmap <- renderPlot({  
 df <- filtered\_data()  
 ggplot(df, aes(hour, date, fill = n)) +  
 geom\_tile(color = "white") +  
 scale\_x\_continuous(breaks = seq(input$hour\_range[1], input$hour\_range[2], by = 1),  
 expand = c(0,0)) +  
 scale\_y\_date(  
 breaks = seq(min(heatmap\_data$date), max(heatmap\_data$date), by = "1 day"),  
 date\_labels = "%b %d",  
 expand = c(0, 0)  
 ) +  
 scale\_fill\_distiller(name = "Messages", palette = "Spectral", direction = 1) +  
 theme\_minimal() +  
 theme(panel.grid = element\_blank())  
 }, res = 96)  
   
   
 # 1.2 Week Comparison  
 output$week\_plot <- renderPlot({  
 df <- week\_patterns %>%  
 filter(week %in% input$weeks,  
 hour >= input$hour\_range2[1],  
 hour <= input$hour\_range2[2])  
 ggplot(df, aes(hour, proportion, color=week)) +  
 geom\_line() + geom\_point() +  
 scale\_y\_continuous(labels=percent\_format()) +  
 theme\_light() +  
 theme(legend.position="top")  
 })  
   
 # 1.3 Influence Bar  
 output$inf\_plot <- renderPlotly({  
 df <- top\_inf %>%  
 filter(target\_subtype == input$subtype)  
 p <- ggplot(df,  
 aes(reorder(sender, n), n,  
 fill = receiver,  
 text = paste0("Sender: ", sender, "\nCount: ", n))) +  
 geom\_col() + coord\_flip() +  
 labs(x="Sender", y="Count") +  
 theme\_minimal()  
 ggplotly(p, tooltip="text")  
 })  
   
 # 1.3 Influence Network  
 output$net\_plot <- renderPlot({  
 req(input$target)  
 root <- which(V(g\_ig)$name == input$target)  
 vids <- unique(c(  
 root,  
 neighbors(g\_ig, root, mode="in"),  
 neighbors(g\_ig, root, mode="out")  
 ))  
 subg <- induced\_subgraph(g\_ig, vids)  
 ggraph(subg, layout="kk") +  
 geom\_edge\_link(color="grey70") +  
 geom\_node\_point(aes(filter=(name==input$target)),  
 color="firebrick", size=6) +  
 geom\_node\_point(aes(filter=(name!=input$target)),  
 color="steelblue", size=4) +  
 geom\_node\_text(aes(label=name),  
 repel=TRUE, size=3) +  
 theme\_void()  
 })  
   
 # ---- Task 2 prep ----  
 g\_undir <- as\_undirected(g\_ig, mode="collapse")  
   
 comm\_obj <- reactive({ cluster\_walktrap(g\_undir) })  
   
 # Shared timeline reactive  
 entity\_meta <- nodes\_tbl %>%  
 filter(type=="Entity") %>%  
 transmute(id,name,sub\_type)  
 timeline\_df <- reactive({  
 msgs %>%  
 left\_join(entity\_meta, by=c("sender"="id")) %>%  
 filter(sub\_type %in% c("Person","Vessel")) %>%  
 mutate(Type=sub\_type)  
 })  
   
 # 2.a Interactive Timeline  
 output$pv\_timeline <- renderPlotly({  
 df <- timeline\_df()  
 p <- ggplot(df,  
 aes(date, hour,  
 color = name,  
 shape = Type,  
 text = paste(name, Type, hour))) +  
 geom\_point(size=3, alpha=0.8) +  
 facet\_wrap(~Type, ncol=1) +  
 scale\_shape\_manual(values=c(Person=16, Vessel=17)) +  
 labs(x="Date", y="Time of Day", color="Sender") +  
 theme\_minimal()  
 ggplotly(p, tooltip="text")  
 })  
 output$task2a\_findings <- renderText({  
 "Findings on People/ Vessels:  
Core logic:  
  
-If two names appear as sender and recipient in the same message, they cannot belong to the same person — i.e., they’re not aliases of each other.  
  
-If two names sent a message at the exact time, they cannot belong to the same person.  
  
-For instance, if Nadia sent a message to The Accountant, they would not be the same individual. If Nadia sent a message at 10am to The Accountant and The Lookout also sent a message at 10am to The Intern, Nadia and The Lookout cannot be the same person.  
  
Select only The Accountant, Mrs. Money, Elise: We see close timings between Mrs. Money and Elise on 8 Oct, and 10 Oct. These were on the same topic. Elise then disappears from radar on 10 Oct. She reappears as The Accountant and Mrs. Money on 11 Oct on the same topic and remains only as The Accountant till 14 Oct.  
Select only Liam and The Middleman: The Middleman disappeared on 7 Oct and appeared as Liam on 8 Oct. On 11 Oct Mrs. Money asked The Middleman if anything was found by conservation vessels. On the same day, Liam reappeared and replied Elise that nothing was found by them.  
  
Select only The Boss and Nadia: The Boss disappeared on 5 Oct and reappeared as Nadia on 8 Oct. Likely the same person.  
  
Select only Small Fry and Rodriguez: on 2 Oct Rodriguez corresponded with Remora and Mako on meeting at the slip #14. It happened again on 14 Oct as he took on dual roles and responded to the same message with different names. Likely the same person.  
  
Select only The Lookout and Sam: on 7 Oct Sam asked Kelly to get information on who authorized the permit. 2 minutes later, The Lookout (Kelly) responded to The Intern (Sam), that it was signed by Jensen from City Council.  
  
Seawatch only appeared on 10 Oct but Horizon talked to Seawatch on 8 Oct. Therefore, some other entity is Seawatch before or during 8 Oct. Defender told Seawatch on 3 Oct at 8.39am that it increased its patrol and informed Seawatch to maintain vigilance. The Lookout (Seawatch) responded to Sentinel (Defender) at 8.41am that it acknowledged the need for vigilance.  
  
 "  
 })  
   
 # 2.b Static Community Detection  
 output$comm\_net\_static <- renderPlot({  
 mem <- membership(comm\_obj())  
 ggraph(g\_undir, layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(color=factor(mem[name])), size=4) +  
 geom\_node\_text(aes(label=name, color=factor(mem[name])),  
 repel=TRUE, size=3) +  
 scale\_color\_manual(values=c("1"="#E41A1C","2"="#377EB8",  
 "3"="#4DAF4A","4"="#984EA3","5"="#FF7F00")) +  
 labs(color="Community") +  
 theme\_void()  
 })  
   
 # --- Before these outputs, define a shared reactive timeline\_df() ---  
 timeline\_df <- reactive({  
 msgs %>%  
 left\_join(  
 nodes\_tbl %>% filter(type=="Entity") %>% transmute(id,name,sub\_type),  
 by = c("sender" = "id")  
 ) %>%  
 filter(sub\_type %in% c("Person","Vessel")) %>%  
 mutate(Type = sub\_type)  
 })  
   
 # 1) Static community‐detection plot  
 output$comm\_graph <- renderPlot({  
 com <- cluster\_louvain(as.undirected(g\_ig, mode="collapse"))  
 mem <- membership(com)  
 ggraph(as.undirected(g\_ig,mode="collapse"), layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(color=factor(mem[name])), size=4) +  
 geom\_node\_text(aes(label=name, color=factor(mem[name])),  
 repel=TRUE, size=3) +  
 scale\_color\_manual(values=c("1"="#E41A1C","2"="#377EB8","3"="#4DAF4A",  
 "4"="#984EA3","5"="#FF7F00")) +  
 theme\_void()  
 })  
   
 # precompute PageRank  
 pr\_vals <- page\_rank(as.undirected(g\_ig,mode="collapse"))$vector  
   
 # 2a) Bar chart of top PageRank nodes  
 output$pr\_bar <- renderPlot({  
 pr\_df <- tibble(  
 name = V(g\_ig)$name,  
 pagerank = pr\_vals  
 ) %>%  
 arrange(desc(pagerank)) %>%  
 slice\_head(n=10)  
   
 ggplot(pr\_df, aes(reorder(name, pagerank), pagerank)) +  
 geom\_col(fill="#377EB8") +  
 coord\_flip() +  
 labs(x="Node", y="PageRank", title="Top 10 by PageRank") +  
 theme\_minimal(base\_size=12)  
 })  
   
 # 2b) Static network sized by PageRank  
 output$pr\_graph <- renderPlot({  
 mem <- membership(cluster\_louvain(as.undirected(g\_ig,mode="collapse")))  
 ggraph(as.undirected(g\_ig,mode="collapse"), layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(size=pr\_vals, color=factor(mem[name])), show.legend="point") +  
 geom\_node\_text(aes(label=name, color=factor(mem[name])),  
 repel=TRUE, size=3) +  
 scale\_size(range=c(2,8), name="PageRank") +  
 scale\_color\_manual(values=c("1"="#E41A1C","2"="#377EB8","3"="#4DAF4A",  
 "4"="#984EA3","5"="#FF7F00")) +  
 theme\_void()  
 })  
   
 # 3) Findings for PageRank step  
 output$task2b\_pr\_findings <- renderText({  
 "Findings:  
- There were 5 closely associated groups. Community 5 (Clepper and Miranda) appeared to be segmented from the central group, due to the non-involvement from the nature of their investigative work.  
From the graph, we extracted the 8 influential nodes to focus on:  
Community 1: Mako  
Community 2: Neptune, Remora, Nadia, Davis  
Community 3: N/A as they were not very influential at global level  
Community 4: Mrs. Money, Boss, The Middleman  
Community 5: N/A as they were not very influential at global level  
Community 6: N/A as they were not very influential at global level"  
 })  
   
 # 4) Membership & top topics (top 5 groups)  
 output$comm\_membership <- renderTable({  
 com <- cluster\_louvain(as.undirected(g\_ig,mode="collapse"))  
 mem <- membership(com)  
 # build membership + topics as before  
 mem\_df <- tibble(name=names(mem), community=mem)  
 part <- edges\_tbl %>%  
 filter(type %in% c("sent","received")) %>%  
 mutate(entity = ifelse(type=="sent", source, target),  
 event = ifelse(type=="sent", target, source)) %>%  
 select(entity, event)  
 about <- edges\_tbl %>% filter(type=="about") %>% transmute(event=source, topic=target)  
 topics <- nodes\_tbl %>% filter(type=="Topic") %>% transmute(id, topic\_name=name)  
   
 joined <- part %>%  
 inner\_join(about, by="event") %>%  
 inner\_join(topics, by=c("topic"="id")) %>%  
 left\_join(mem\_df, by=c("entity"="name"))  
   
 top\_topics <- joined %>%  
 count(community, topic\_name, name="n") %>%  
 group\_by(community) %>%  
 slice\_max(n, n=3) %>%  
 summarise(predominant = paste(topic\_name, collapse=", "))  
   
 members <- mem\_df %>%  
 group\_by(community) %>%  
 summarise(Members=paste(name, collapse=", "))  
   
 out <- members %>%  
 left\_join(top\_topics, by="community") %>%  
 rename(Group=community) %>%  
 arrange(Group) %>%  
 slice\_head(n=5)  
   
 out  
 }, bordered=TRUE, spacing="xs")  
   
  
   
 # 6) Interactive community timeline  
 output$comm\_timeline <- renderPlotly({  
 # reuse timeline\_df() defined above Task 2.a  
 df <- timeline\_df() %>%  
 left\_join(tibble(name=names(membership(cluster\_louvain(as.undirected(g\_ig,mode="collapse")))),  
 community=membership(cluster\_louvain(as.undirected(g\_ig,mode="collapse")))),  
 by="name")  
 p <- ggplot(df, aes(date, hour, color=factor(community), shape=Type,  
 text=paste(name, "Grp", community, hour))) +  
 geom\_point(size=3, alpha=0.8) +  
 facet\_wrap(~Type, ncol=1) +  
 scale\_shape\_manual(values=c(Person=16, Vessel=17)) +  
 labs(x="Date", y="Time of Day", color="Community") +  
 theme\_minimal()  
 ggplotly(p, tooltip="text")  
 })  
   
 # 7) Discussion / Interpretation  
 output$task2b\_discussion <- renderText({  
 "Discussion:  
We mainly focused on the conversations by 8 influential nodes and some related nodes:  
Conservation Group (Comm.1): Samantha Blake informed Mako to stop operations on 8 and 10th Oct. Serenity is a private luxury yacht. Osprey was likely a tourism vessel looking for charter from Mako for their tourists.  
  
Permit (Comm. 2): Neptune, Remora, Nadia, and Davis were working on Nemo Reef operation. This referred to the Music Video Production for Sailor Shift on 14 Oct.  
  
Pseudonym (Comm. 3): Other than communicating among themselves, The Lookout appeared to have also externally corresponded with Sentinel, Reef Guardian and Horizon (conservation based topics), while The Intern also externally corresponded with Mrs. Money.  
  
Suspicious (Comm. 4): The Middleman had access to Council documents. Mrs. Money had funding from sources that would not flag out to regulators for her operations. Mrs. Money was investigating V. Miesel’s structures. On 5 Oct, Boss told Mrs. Money to disguise financial trails through tourism ventures and destroy evidence of Nemo Reef operations.  
  
Hacklee Herald (Comm. 5): Conversations between Clepper and his intern Miranda which ended on 11 Oct. Miranda mentioned an Oceanus City Council Member meeting with unmarked vessels at night."  
 })  
   
   
 # --- Task 3: rule-based pseudonym detection ----  
   
 # 3.1 Build the pseudo table  
 nodes\_person <- nodes\_tbl %>%  
 filter(sub\_type == "Person") %>%  
 transmute(id, label = name, sub\_type)  
   
 pseudonym\_keywords <- c("Boss", "Accountant", "Middleman", "Lookout",  
 "Intern", "Money", "Fry")  
   
 pseudo\_df <- nodes\_person %>%  
 mutate(  
 is\_prefix = str\_detect(label, "^The "),  
 is\_keyword = str\_detect(label, paste(pseudonym\_keywords, collapse = "|")),  
 is\_pseudonym = is\_prefix | is\_keyword  
 ) %>%  
 filter(is\_pseudonym) %>%  
 select(id, label, sub\_type)  
   
 output$pseudo\_table <- renderTable({  
 pseudo\_df  
 }, striped = TRUE, bordered = TRUE)  
   
 # 3.2 Build & render the pseudonym-only communication network  
 pseudo\_labels <- pseudo\_df$label  
   
 pv\_nodes <- data.frame(id = pseudo\_labels, label = pseudo\_labels)  
   
 pv\_edges <- msgs %>%  
 # attach sender label  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("sender" = "id")) %>%  
 rename(from = label) %>%  
 # attach receiver label  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("receiver" = "id")) %>%  
 rename(to = label) %>%  
 filter(from %in% pseudo\_labels, to %in% pseudo\_labels) %>%  
 distinct(from, to)  
   
 output$pseudonym\_net <- renderVisNetwork({  
 visNetwork(pv\_nodes, pv\_edges) %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)  
 })  
   
 # 3.3 Conclusion  
 output$task3\_conclusion <- renderText({  
 paste(  
 "→ Identified pseudonyms:",  
 paste(pseudo\_df$label, collapse = ", "),  
 "",  
 "With the explicit identification of pseudonyms such as “Boss,” “The Intern,” “The Accountant,” “Mrs. Money,” “The Middleman,” “The Lookout,” and “Small Fry,” my understanding of the activities within the Oceanus network changed significantly.",  
 "Initially, communications and relationships appeared to be fragmented among many individuals and vessels. However, after mapping pseudonyms onto the network, several important patterns emerged:",  
 "",  
 "-Pseudonyms as Key Hubs: In the visual network, the detected pseudonyms are not peripheral actors—they consistently occupy central hub positions, connecting to multiple vessels and people through various types of relationships (such as “Reports,” “Coordinates,” “Operates,” and “Suspicious”).",  
 "-Obscured True Roles: What previously looked like many unrelated participants was actually a small set of influential actors operating under multiple pseudonyms. For example, “The Intern” and “Boss” are both connected to several vessels via “Coordinates” and “Operates” relationships, indicating they may be orchestrating or managing operations behind the scenes.).",  
 "-Concealment of Illicit Activity: The structure revealed that pseudonyms are often used to mask direct links between real individuals and vessels—particularly in suspicious or high-frequency communication events. This suggests intentional obfuscation, possibly to hide illicit coordination or the flow of sensitive information.)",  
 "-Redefining Influence and Risk: By highlighting pseudonyms, it becomes clear that influence within the network is far more centralized than the real-name view suggested. A few pseudonyms control or interact with a large proportion of vessels and people, making them high-value targets for further investigation.",  
 sep = "\n")  
 })  
   
   
 # --- Task 4: Nadia Conti Investigation ---  
   
 ## ——————— SERVER ———————  
   
 # 4.0 Nadia’s node ID  
 nadia\_id <- nodes\_tbl %>% filter(name == "Nadia Conti") %>% pull(id)  
   
 # — 4.1.1 Nadia’s 1-Hop Ego Network (Louvain) —  
 ego1 <- make\_ego\_graph(as.undirected(g\_ig, mode="collapse"),  
 order = 1,  
 nodes = which(V(g\_ig)$name == "Nadia Conti"))[[1]]  
 com1 <- cluster\_louvain(ego1)  
 mem1 <- membership(com1)  
   
 output$nadia\_ego\_1hop <- renderPlot({  
 ggraph(ego1, layout="kk") +  
 geom\_edge\_link(color="grey80", alpha=0.5) +  
 geom\_node\_point(aes(color = factor(mem1)), size=5) +  
 geom\_node\_text(aes(label = name, color = factor(mem1)),  
 repel=TRUE) +  
 labs(title = "Nadia’s 1-Hop Ego Network", color = "Comm") +  
 theme\_void()  
 })  
   
 # — 4.1.2 Sent vs Received Communications Networks —  
   
 # prepare sent edges/nodes  
 sent\_edges1 <- msgs %>%  
 filter(sender == nadia\_id) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("sender" = "id")) %>% rename(from = label) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("receiver" = "id")) %>% rename(to = label) %>%  
 distinct(from, to)  
 sent\_nodes1 <- tibble(id = unique(c(sent\_edges1$from, sent\_edges1$to)),  
 label = id)  
   
 output$nadia\_sent\_net <- renderVisNetwork({  
 g\_sent <- graph\_from\_data\_frame(sent\_edges1,  
 vertices = sent\_nodes1,  
 directed = TRUE)  
 visIgraph(g\_sent, layout = "layout\_with\_fr") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)  
 })  
   
 # prepare received edges/nodes  
 recv\_edges1 <- msgs %>%  
 filter(receiver == nadia\_id) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("sender" = "id")) %>% rename(from = label) %>%  
 left\_join(nodes\_tbl %>% transmute(id, label = name),  
 by = c("receiver" = "id")) %>% rename(to = label) %>%  
 distinct(from, to)  
 recv\_nodes1 <- tibble(id = unique(c(recv\_edges1$from, recv\_edges1$to)),  
 label = id)  
   
 output$nadia\_recv\_net <- renderVisNetwork({  
 g\_recv <- graph\_from\_data\_frame(recv\_edges1,  
 vertices = recv\_nodes1,  
 directed = TRUE)  
 visIgraph(g\_recv, layout = "layout\_with\_fr") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE)  
 })  
   
 # — 4.1.3 Findings —  
 output$task4\_1\_findings <- renderText({  
 paste0(  
 "Nodes involved: Nadia, Davis, Elise, Haacklee Harbor, Liam, Marlin,\n",  
 "Neptune, Oceanus City Council, Remora, Rodriguez,\n",  
 "Sailor Shifts Team, Sentinel, V. Miesel Shipping.\n\n",  
 "• Sent more direct communications to Liam (2) and Neptune (2).\n",  
 "• Received more communications from Davis (5), Elise (3), Liam (2).\n\n",  
 "Next step: drill into suspicious chars/activities."  
 )  
 })  
   
 # — 4.1.4 Nadia’s 2-Hop Ego (Degree Centrality) —  
 ego2 <- make\_ego\_graph(as.undirected(g\_ig, mode="collapse"),  
 order = 2,  
 nodes = which(V(g\_ig)$name == "Nadia Conti"))[[1]]  
 deg2 <- degree(ego2)  
 deg2\_df <- tibble(name = names(deg2), degree = deg2)  
   
 output$nadia\_ego\_2hop <- renderPlot({  
 ggplot(deg2\_df, aes(reorder(name, degree), degree)) +  
 geom\_col() +  
 coord\_flip() +  
 labs(title = "Nadia’s 2-Hop Ego (Degree Centrality)",  
 x = "Node", y = "Degree") +  
 theme\_minimal()  
 })  
   
 # — 4.1.5 Conclusion —  
 output$task4\_1\_conclusion <- renderText({  
 paste0(  
 "1-Hop nodes: Nadia, Davis, Elise, Haacklee Harbor, Liam, Marlin,\n",  
 "Neptune, Oceanus City Council, Remora, Rodriguez,\n",  
 "Sailor Shifts Team, Sentinel, V. Miesel Shipping.\n\n",  
 "2-Hop additions: EcoVigil, Sam, The Accountant, Nemo Reef.\n\n",  
 "Use these 2-hop nodes to drill deeper for suspicious links."  
 )  
 })  
   
 # — 4.2.1 Plotted Timeline for all 2-Hop actors —  
 hop2\_names <- V(ego2)$name  
 id\_map <- nodes\_tbl %>% filter(name %in% hop2\_names) %>% transmute(id, name)  
 timeline2 <- msgs %>%  
 filter(sender %in% id\_map$id | receiver %in% id\_map$id) %>%  
 mutate(Actor = if\_else(sender %in% id\_map$id,  
 id\_map$name[match(sender, id\_map$id)],  
 id\_map$name[match(receiver, id\_map$id)]))  
   
 output$nadia\_2hop\_timeline <- renderPlotly({  
 p <- ggplot(timeline2, aes(date, hour, color = Actor,  
 text = paste0(Actor, "\n", format(ts, "%Y-%m-%d %H:%M")))) +  
 geom\_point(size = 3, alpha = 0.8) +  
 labs(x = "Date", y = "Hour", color = "Actor") +  
 theme\_minimal()  
 ggplotly(p, tooltip = "text")  
 })  
   
 # — 4.2.2 Ego & Comms Table for 2-Hop actors —  
 comm2\_table <- timeline2 %>%  
 transmute(  
 Date = as.character(date),  
 Time = format(ts, "%H:%M"),  
 Sender = nodes\_tbl$name[match(sender, nodes\_tbl$id)],  
 Receiver = nodes\_tbl$name[match(receiver, nodes\_tbl$id)],  
 EventID = event\_id  
 )  
   
 output$nadia\_comm\_table <- DT::renderDataTable({  
 DT::datatable(comm2\_table,  
 options = list(pageLength = 10, autoWidth = TRUE),  
 rownames = FALSE)  
 })  
   
   
   
   
   
}  
  
  
shinyApp(ui, server)