```
library(readr)      # For reading CSV files
df <- read_csv("job_fraud (1).csv")
```

```
## Rows: 17879 Columns: 18
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (13): title, location, department, salary_range, company_profile, descri...
## dbl  (5): job_id, telecommuting, has_company_logo, has_questions, fraudulent
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
df_cleaned <- df %>%
  distinct() %>%                          # Remove duplicate rows
  filter(!is.na(description)) %>%         # Remove rows with missing descriptions
  mutate(across(
    c(salary_range, department, required_education, benefits,
      required_experience, `function`, industry, employment_type,
      company_profile, requirements, location),
    ~ replace_na(., "Unknown")            # Replace NA with "Unknown"
  ))
```

```
library(knitr)

df_cleaned %>%
  group_by(employment_type, fraudulent) %>%
  summarise(Count = n(), .groups = "drop") %>%
  pivot_wider(names_from = fraudulent, values_from = Count, values_fill = 0) %>%
  rename(`Real Jobs` = `0`, `Fake Jobs` = `1`) %>%
  kable()
```

employment_type	Real Jobs	Fake Jobs
Contract	1480	44
Full-time	11130	490
Other	212	15
Part-time	723	74
Temporary	239	2
Unknown	3230	240

```
library(dplyr)
library(tidyr)
library(knitr)

df_cleaned %>%
  group_by(required_experience, fraudulent) %>%
  summarise(Count = n(), .groups = "drop") %>%
  pivot_wider(names_from = fraudulent, values_from = Count, values_fill = 0) %>%
  rename(`Real Jobs` = `0`, `Fake Jobs` = `1`) %>%
  arrange(desc(`Fake Jobs`)) %>%
  kable()
```

required_experience	Real Jobs	Fake Jobs
Unknown	6615	434
Entry level	2518	179
Mid-Senior level	3696	113
Not Applicable	1056	60
Associate	2255	42
Director	372	17
Executive	131	10
Internship	371	10

```
df_cleaned %>%
  group_by(employment_type) %>%
  summarise(
    Total = n(),
    Fake = sum(fraudulent),
    `Fraud %` = round(100 * mean(fraudulent), 2)
  ) %>%
  arrange(desc(`Fraud %`)) %>%
  kable()
```

employment_type	Total	Fake	Fraud %
Part-time	797	74	9.28
Unknown	3470	240	6.92
Other	227	15	6.61
Full-time	11620	490	4.22
Contract	1524	44	2.89
Temporary	241	2	0.83

```
library(shiny)
library(ggplot2)
library(dplyr)
library(bslib)
```

```
##
## Attaching package: 'bslib'
```

```
## The following object is masked from 'package:utils':
##
##     page
```

```r
# Load data
df_cleaned <- read.csv("job_fraud (1).csv")
df_cleaned$fraudulent <- factor(df_cleaned$fraudulent, labels = c("Real", "Fake"))

# UI
ui <- fluidPage(
  theme = bs_theme(bootswatch = "darkly"),

  tags$head(
    tags$style(HTML("
      h1 {
        text-align: center;
        font-weight: bold;
        margin-bottom: 20px;
      }
      .select-row {
        display: flex;
        justify-content: center;
        gap: 30px;
        margin-bottom: 30px;
      }
    "))
  ),

  h1("Job Postings Fraud Detection Dashboard"),

  div(class = "select-row",
    selectInput("category", "Select Category:",
                choices = c("industry", "employment_type", "required_experience"),
                selected = "employment_type"),

    selectInput("job_type", "Filter by Job Type:",
                choices = c("All", "Real", "Fake"),
                selected = "All"),

    selectInput("top_n", "Show Top N Categories:",
                choices = c(5, 10, 15),
                selected = 10)
  ),

  plotOutput("barPlot", height = "550px")
)

# Server
server <- function(input, output) {
  output$barPlot <- renderPlot({
    col_name <- input$category
    top_n_val <- as.numeric(input$top_n)

    top_categories <- df_cleaned %>%
      count(.data[[col_name]]) %>%
      arrange(desc(n)) %>%
      slice_head(n = top_n_val) %>%
      pull(1)

    data_filtered <- df_cleaned %>%
      filter(.data[[col_name]] %in% top_categories)

    if (input$job_type != "All") {
      data_filtered <- data_filtered %>%
        filter(fraudulent == input$job_type)
    }

    ggplot(data_filtered, aes_string(x = col_name, fill = "fraudulent")) +
      geom_bar(position = "dodge", alpha = 0.85) +
      scale_fill_manual(values = c("Real" = "#3498db", "Fake" = "#e74c3c")) +
      labs(
        title = paste("Distribution of", input$job_type, "Jobs by", col_name),
        x = col_name, y = "Number of Postings", fill = "Job Type"
      ) +
      theme_minimal(base_size = 14) +
      theme(
        plot.title = element_text(size = 18, face = "bold", color = "orange", hjust = 0.5),
        axis.text.x = element_text(angle = 30, hjust = 1, color = "white"),
        axis.text.y = element_text(color = "white"),
        axis.title = element_text(color = "white"),
        panel.grid.major = element_line(color = "grey40"),
        panel.grid.minor = element_line(color = "grey25"),
        plot.background = element_rect(fill = "#222222", color = NA),
        panel.background = element_rect(fill = "#222222"),
        legend.background = element_rect(fill = "#222222"),
        legend.title = element_text(color = "white"),
        legend.text = element_text(color = "white"),
        legend.position = "top"
      )
  })
}

# Run the app
shinyApp(ui = ui, server = server)
```