K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY
(AUTONOMOUS), TRICHY

CREATING BASIC MAPS TO DISPLAY POPULATION DENSITY OF INDIAN STATES

PRESENTED BY

2303811724321067-MOHAMED FIRDOUS S

PRESENTATION OVERVIEW

- Problem Identification and Analysis
- Objective
- Proposed Work Architecture & Module Design
- Block diagram of proposed system
- Module Description
- Module Implementation
- R Programming Implementation
- Source Code
- Output Snapshots
- Conclusion

PROBLEM IDENTIFICATION

Problem Identification

•Population data from sources like the Census is vast and difficult to interpret manually.

•Users lack intuitive tools to explore population density by state interactively.

•Traditional tabular formats make it hard to identify high or low-density regions at a glance.

Problem Analysis

•There is a need for a dynamic, map-based visualization platform for population data.

•Interactive features like filtering and color-coded density maps can enhance data exploration.

•A web-based interface using R Shiny can make data accessible and understandable for all users.

OBJECTIVE

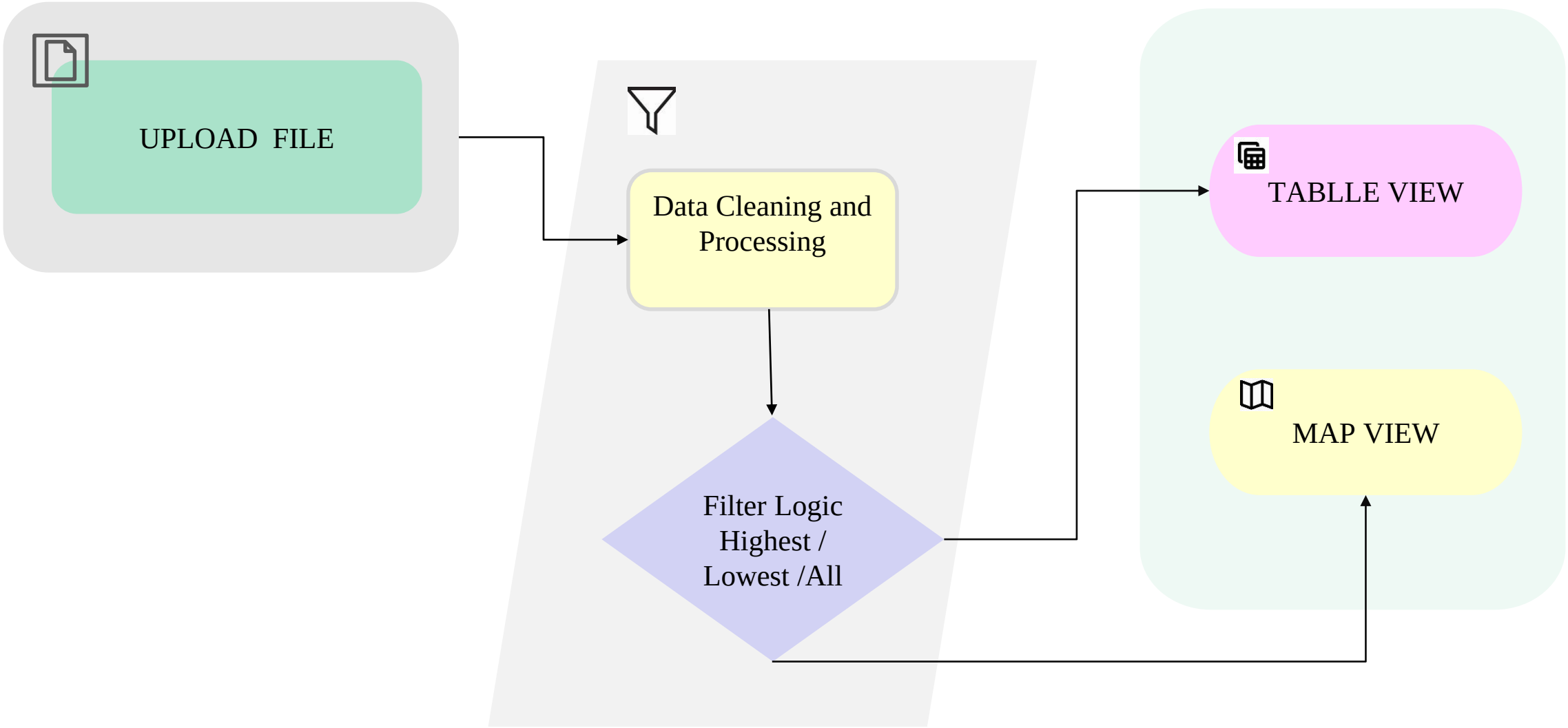
To develop a user-friendly dashboard that:

- ❑ Displays population density data for Indian states.
- ❑ Provides both tabular and visual (map-based) views.
- ❑ Supports CSV file uploads and filtering by density level

Proposed Work Architecture & Module Design

- **Frontend:** Built using **R Shiny** and **shinydashboard** to provide an intuitive and interactive user interface.
- **Backend:** Processes uploaded data and user interactions using **R functions** and **dplyr** for data transformation.
- **Visualization:** Implements **Leaflet** for map visualization and **DT** for dynamic table display of population data.
- **Modules:** Includes **Data Upload & Cleaning**, **Table View**, **Map Visualization**, **User Interface**, and **Integration & Testing**.
- **Security & Deployment:** Supports basic session handling and is deployable via **shinyapps.io** or local server environments.

BLOCK DIAGRAM



MODULES DESCRIPTION

- Data Upload & Cleaning Module
- Table View Module
- Map Visualization Module
- User Interface Design using Shiny
- Integration and Testing Module

MODULES IMPLEMENTATION

Data Upload & Cleaning Module

- Uploads .csv files containing population and state data
- Parses string numbers with commas into numeric format
- Merges data if multiple datasets are used
- Prepares clean dataset for visualization
- Cleans data by removing missing or invalid entries

MODULES IMPLEMENTATION

Table View Module

- Presents cleaned dataset in an interactive table
- Allows sorting and searching by state or density
- Supports pagination for large datasets
- Reflects filtering choices from the user
- Makes comparison and data interpretation easier

MODULES IMPLEMENTATION

Map Visualization Module

- Displays population density data using interactive maps
- Uses color-coded markers to represent density levels
- Tooltips show state name and population details
- Zoom and pan for exploring different regions
- Highlights patterns and outliers geographically

MODULES IMPLEMENTATION

User Interface Design using Shiny

- Built using R Shiny’s reactive UI components
- Sidebar includes controls for upload, filter, and view selection
- Tabs for toggling between Table and Map views
- Responsive layout for desktop and mobile
- Clean, user-friendly interface for smooth navigation

MODULES IMPLEMENTATION

Integration and Testing Module

- Combines all modules into a fully functional Shiny app
- Validates input handling and reactive updates
- Tests usability across devices and browsers
- Debugs and optimizes UI and server logic
- Ensures smooth operation before final deployment

DATA SCIENCE CONCEPTS

- **Shiny** – For interactive UI and server logic.
- **leaflet** – For dynamic geographic visualization.
- **DT** – For interactive and filterable tables.
- **dplyr** – For data filtering and manipulation.

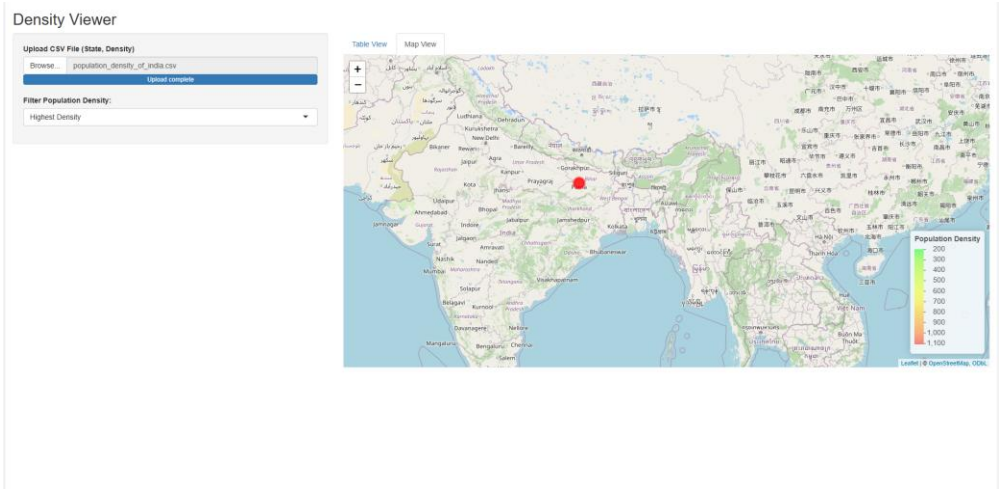
SOURCE CODE

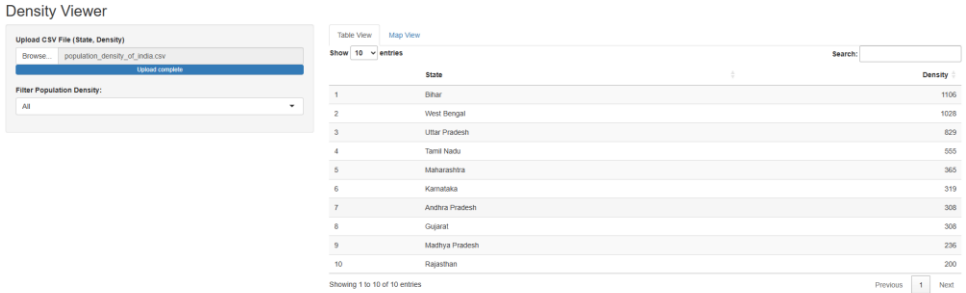
```
library(shiny)
library(DT)
library(leaflet)
library(dplyr)

state_coords <- data.frame(
  State = c("Uttar Pradesh", "Bihar", "Maharashtra", "West Bengal", "Tamil
Nadu",
            "Rajasthan", "Karnataka", "Gujarat", "Andhra Pradesh", "Madhya
Pradesh"),
  Lat = c(26.85, 25.59, 19.75, 22.57, 11.12, 27.02, 15.31, 22.26, 15.91,
23.52),
  Lon = c(80.91, 85.13, 75.71, 88.36, 78.15, 74.22, 75.71, 72.57, 79.74,
77.81)
)

ui <- fluidPage(
  titlePanel("Density Viewer"),

  sidebarLayout(
    sidebarPanel(
      fileInput("file", "Upload CSV File (State, Density)", accept = ".csv"),
      selectInput("filter_type", "Filter Population Density:",
                  choices = c("All", "Highest Density", "Lowest Density"))
    ),

    mainPanel(
      tabsetPanel(
        tabPanel("Table View", DTOutput("density_table")),
        tabPanel("Map View", leafletOutput("density_map", height = 600))
      )
    )
  )
)

server <- function(input, output) {

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

      df$Density <- as.numeric(gsub(",", "", df$Density))

       merged <- df %>%
      inner_join(state_coords, by = "State")

    merged
  })
```

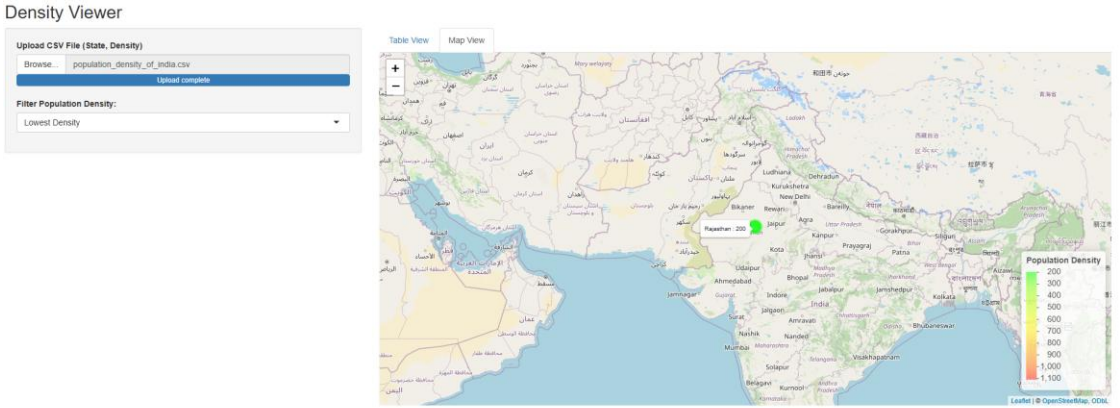
SOURCE CODE

```
filtered_data <- reactive({
    data <- user_data()
    if (input$filter_type == "Highest Density") {
      data %>% filter(Density == max(Density, na.rm = TRUE))
    } else if (input$filter_type == "Lowest Density") {
      data %>% filter(Density == min(Density, na.rm = TRUE))
    } else {
      data
    }
  })
output$density_table <- renderDT({
    datatable(filtered_data()[, c("State", "Density")], options = list(pageLength
= 10))
  })

  output$density_map <- renderLeaflet({
    req(filtered_data())
    pal <- colorNumeric(palette = c("green", "yellow", "red"), domain =
user_data()$Density)
    leaflet() %>%
      addTiles() %>%
      addCircleMarkers(data = filtered_data(),
                       lat = ~Lat,
                       lng = ~Lon,
                       radius = 10,
                       color = ~pal(Density),
                       stroke = TRUE,
                       fillOpacity = 0.8,
                       label = ~paste(State, ": ", Density)) %>%
      addLegend("bottomright", pal = pal, values = user_data()$Density,
              title = "Population Density")
  })
}

shinyApp(ui = ui, server = server)
```

Output Snapshots







Output Snapshots

Density Viewer

Upload CSV File (State, Density)

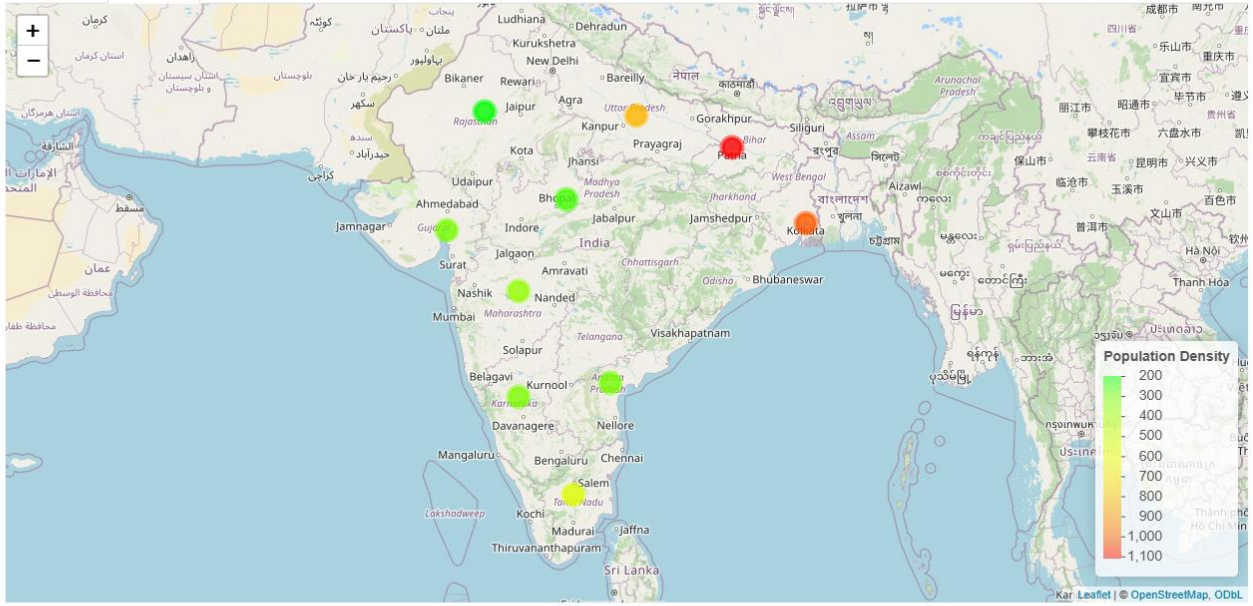
Browse... population_density_of_india.csv

Upload complete

Filter Population Density:

All

Table View | Map View



CONCLUSION

- Successfully built a dynamic “Density Viewer” dashboard using R.
- Helps visualize and analyze population density across Indian states.
- Easy to use and extendable for other geospatial datasets.

Future scope includes:

- **Role-based Access System**: Enable different user roles such as administrators, researchers, and public users with tailored access and functionality.
- **Downloadable Reports**: Allow users to export filtered data and visualizations in formats like PDF, CSV, or PNG for offline analysis.

THANK YOU