Forecasting Next Week's Temperatures Using a Simple Moving Average

A PROJECT REPORT

Submitted by

SATHIYAJITH AASHIM GOVINDARAJ (2303811724321098)

in partial fulfillment of requirements for the award of the course

AGI1252 - FUNDAMENTALS OF DATA SCIENCE USING R

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE- 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"Forecasting Next Week's Temperatures Using a Simple Moving Average"** is the bonafide work of **SATHIYAJITH AASHIM GOVINDARAJ (2303811724321098)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.T. AVUDAIAPPAN, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

SIGNATURE

Ms.S.Murugavalli., M.E.,(Ph.D).,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on ……**02.06.2025**.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on **"Forecasting Next Week's Temperatures Using a Simple Moving Average"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **AGI1252 - FUNDAMENTALS OF DATA SCIENCE USING R**

.

Signature

SATHIYAJITH AASHIM GOVINDARAJ

Place: Samayapuram

Date:30.05.2025

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Ms.S.Murugavalli., M.E.,(Ph.D).,** Department of **ARTIFICIAL INTELLIGENCE,** for her incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

- To serve the society by offering top-notch technical education on par with global standards.

Mission:

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

DEPARTMENT

Vision:

- To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfil industrial demands and societal expectations.

Mission

- To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
- To collaborate with industry and offer top-notch facilities in a conducive learning environment.
- To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
- To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
- **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
- **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
- **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply knowledge of mathematics, natural science, computing, engineering fundamentals, and an engineering specialization to develop solutions to complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.
3. **Design/development of solutions:** Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.
4. **Conduct investigations of complex problems:** Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.
5. **Engineering Tool Usage:** Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.
6. **The Engineer and The World:** Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

7. **Ethics:** Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

8. **Individual and Collaborative Team work:** Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

9. **Communication:** Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.

10. **Project management and finance:** Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

11. **Life-long learning:** Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

ABSTRACT

This project presents a user-friendly temperature forecasting application developed using R and the Shiny framework. The main goal is to predict short-term temperature trends using the Simple Moving Average (SMA) method, a common statistical technique that smooths fluctuations in time-series data to identify trends. Users can upload historical temperature data in CSV format, select a location, set a forecast start date, choose the number of forecast days, and adjust the SMA window size. The application processes this input, computes historical trends, and forecasts future temperatures based on the latest SMA value. A synthetic dataset simulating two years of daily temperature data for multiple regions was used to evaluate the tool's performance. The results are displayed on a separate output interface with a color-coded graph showing actual vs. forecasted temperatures, a detailed table, and a smart trend insight indicating whether the future trend is "Rising ," "Dropping ," or "Stable ." The dual-interface design ensures better user interaction and clarity in visualizing results. This lightweight and interactive tool can be applied in fields such as agriculture, education, and climate monitoring. Future improvements may include real-time weather API integration, support for advanced forecasting models, and broader regional coverage.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
This project presents a dynamic temperature forecasting application built with R and Shiny. It enables users to upload historical weather data, select locations, define forecast periods, and apply the Simple Moving Average (SMA) technique to predict short-term temperature trends. The app features a two-interface design: an input panel for data selection and parameters, and an output panel displaying colorful graphs, forecast tables, and trend insights (Rising, Dropping, or Stable). A sample synthetic dataset simulating two years of daily temperatures for various regions is used to demonstrate functionality. The tool is user-friendly, visually intuitive, and adaptable for use in agriculture, education, and climate analysis. Future enhancements may include real-time API integration and advanced forecasting models.	**PO1 -3** **PO2 -3** **PO3 -2** **PO4 -2** **PO5 -1** **PO6 -3** **PO7 -2** **PO8 -1** **PO9 -2** **PO10 -1** **PO11-3** **PO12 -2**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Objective

The objective of this project is to develop an interactive **Temperature Forecasting Application** using the **R Shiny framework**. The application allows users to upload CSV files containing historical temperature data and apply **Simple Moving Average (SMA)** techniques to forecast temperatures for upcoming days. The tool helps users visualize trends and patterns using dynamic plots and tabular representations, providing insightful analysis based on location-wise data.

1.2 Overview

This project provides a user-friendly two-interface design:

- **Input Interface**: Used to upload a CSV file, select a location, set the forecast start date, specify the number of days to forecast, and choose the SMA window size.
- **Output Interface**: Displays forecasted results through colorful line graphs, a summary insight panel, and forecast tables.

The app processes data in real time, applies statistical forecasting (SMA), and dynamically generates meaningful visualizations and insights. The interface is intuitive, supports CSV uploads, and presents the results in a clean, engaging manner with enhanced UI features like color coding and emojis to indicate trends (e.g., for rising temperature).

1.3 Data Handling and Processing Concepts in R

This project leverages the powerful capabilities of R for reading, processing, and analyzing time-series temperature data. The input is a CSV file containing columns such as Location, Date, and Temperature, which are processed using a combination of R packages including ggplot2.

The key steps in data handling and processing include:

- **Data Import and Cleaning**: The uploaded CSV file is read using read.csv(), and the column names are converted to lowercase for consistency. The Date column is transformed into a proper date format using as.Date(), and the Location field is treated as a categorical variable using as.factor().
- **Filtering and Selection**: Users can select a specific location and define a start date for forecasting. The application filters the dataset accordingly to isolate relevant historical data for analysis.
- **Rolling Average Calculation**: The project uses the rollmean() function to compute the Simple Moving Average (SMA) over a user-defined window. This technique smooths temperature fluctuations and captures recent trends effectively.
- **Forecast Generation**: The most recent SMA value is used as a base to forecast temperatures for the upcoming days. These forecasted values are appended to the historical data for visualization and interpretation.
- **Data Visualization**: The processed data—both actual and forecasted—is visualized using ggplot2. Colorful line charts, trend indicators, and interactive tables present the results in an intuitive and user-friendly format.

All operations are performed within the R Shiny framework, ensuring that the application remains dynamic, responsive, and easy to use. The data handling

pipeline is designed to support real-time interaction, enabling users to experiment with different parameters and instantly view the results.

CHAPTER 2

PROJECT METHODOLOGY

2.1 Proposed Work

The proposed project aims to design and implement a **user-friendly, two-interface temperature forecasting web application** using **R Shiny**. The system allows users to upload historical temperature data in CSV format and visualize future temperature forecasts using the **Simple Moving Average (SMA)** technique.

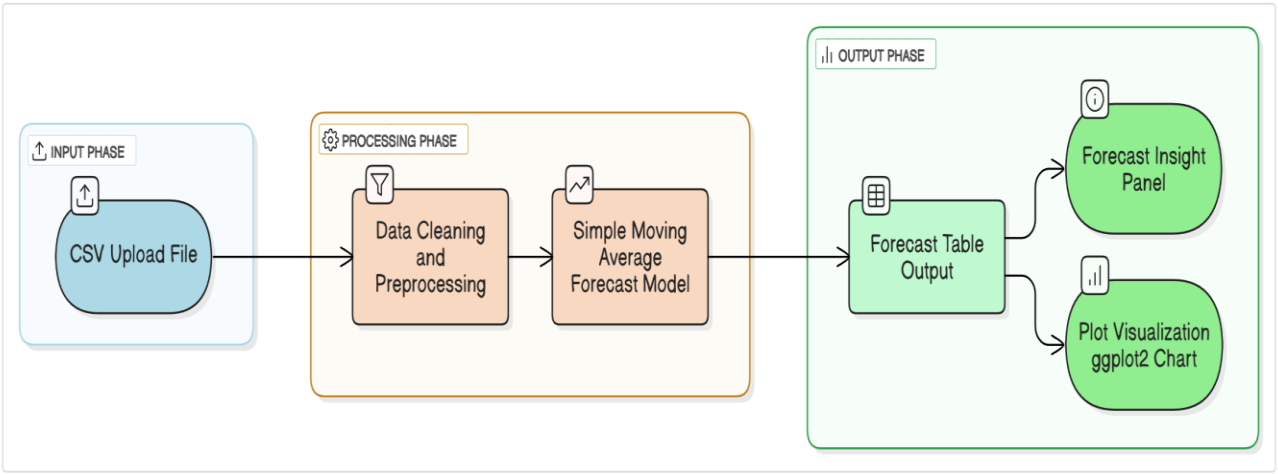
The work is divided into the following key stages:

- **Data Input Module**
  - Users upload a CSV file containing historical data with columns: Location, Date, and Temperature.
  - The application dynamically detects unique locations and date formats and allows users to configure parameters like SMA window and forecast duration.
- **Preprocessing and Transformation**
  - The system converts all date formats into a standard dd/mm/yyyy format for uniformity.
  - Filters and sorts data by location and date.
  - Applies the SMA algorithm on historical data to generate a smoothed value used for forecasting.
- **Forecasting Module**
  - Based on user-defined forecast days and SMA window, the application projects future temperatures.

- Forecast values are generated using the last computed SMA value from the training dataset.

- **Output and Visualization**
  - Results are shown in a separate tab with:
  - A **colorful line plot** comparing historical vs. forecasted temperatures.
  - A **tabular view** of forecasted values.
  - A **forecast insight panel** with emoji-based trend representation (e.g., Rising, Dropping, Stable).

2.2 Block Diagram:



CHAPTER 3

MODULE DESCRIPTION

The proposed Temperature Forecasting Application is divided into **six main modules**, each responsible for a specific functionality. These modules work together seamlessly to provide a smooth user experience and meaningful output.

3.1 File Upload Module

- **Purpose**: To allow users to upload a CSV file containing historical temperature data.
- **Functionality**:
  - Accepts files with columns: Location, Date, and Temperature.
  - Validates the file format and displays error messages if the structure is incorrect.
  - Converts date format to dd/mm/yyyy for uniform display.
- **Tools Used**: fileInput(), read.csv(), as.Date(), dplyr

3.2 Location and Forecast Parameter Selection Module

- **Purpose**: To allow the user to select the **location** and configure the **forecast parameters**.
- **Functionality**:
  - Dynamic dropdown to select available locations.
  - Inputs for:
    - Forecast start date
    - Number of forecast days
    - SMA (Simple Moving Average) window

- Trigger button to start forecasting.

- **Tools Used**: selectInput(), dateInput(), numericInput(), actionButton()

3.3 Data Preprocessing and Transformation Module

- **Purpose**: To clean and prepare the data before forecasting.
- **Functionality**:
  - Sorts data by location and date.
  - Filters data up to the forecast start date.
  - Converts date formats, handles missing values, and calculates SMA values.
- **Tools Used**: dplyr, zoo::rollmean, mutate(), arrange()

3.4 Forecast Generation Module

- **Purpose**: To generate forecasted temperature values using SMA.
- **Functionality**:
  - Computes SMA on historical temperature data.
  - Uses the most recent SMA value to forecast temperatures for the selected number of days.
  - Generates a new dataframe with forecasted values and combines it with historical data.
- **Tools Used**: reactive(), eventReactive(), zoo, seq.Date()

3.5 Output Visualization Module

- **Purpose**: To present forecast results graphically.
- **Functionality**:
  - Displays a **color-coded line chart** showing historical and forecasted data.
  - Red = Forecast, Blue = Actual.

    - Allows users to visually identify trends and changes.
- **Tools Used**: ggplot2, renderPlot(), geom_line(), scale_color_manual()

3.6 Forecast Table and Insight Module

- **Purpose**: To summarize forecast data in table form and provide intelligent insights.
- **Functionality**:
    - Displays forecasted temperatures in a structured table.
    - Computes and compares average values to show a **trend** (Rising , Dropping , or Stable).
    - Insights displayed using dynamic HTML with color and emoji-based representation.
- **Tools Used**: renderTable(), renderUI(), HTML(), mean()

CHAPTER 4

CONCLUSION & FUTURE SCOPE

4.1 Conclusion

This project successfully implements a **Temperature Forecasting Application** using **R Shiny**, enabling users to forecast future temperatures based on historical data through the **Simple Moving Average (SMA)** technique. The application is designed with a clean, colorful, and user-friendly two-interface layout:

- **Input Interface**: Allows file upload, parameter selection (location, date, SMA window, forecast duration).
- **Output Interface**: Displays forecasted results using dynamic plots, data tables, and a visual insight panel with emojis and trends.

Key achievements of the project include:

- Accurate and easy-to-understand temperature forecasting.
- Interactive visualizations with trend interpretation.
- Clean handling of dates in dd/mm/yyyy format.
- Modular code structure for scalability and maintainability.

This project demonstrates the power of **R and Shiny** for building real-time, interactive data applications in a web-based environment, providing meaningful insights to users in a visually engaging way.

4.2 Future Scope

The project lays a strong foundation and opens up several opportunities for future enhancements:

- **Database Integration**
    - Replace CSV upload with a live **MySQL/PostgreSQL** database connection to manage and query large datasets efficiently.
- **Advanced Forecasting Models**
    - Incorporate **ARIMA**, **Exponential Smoothing**, or **Machine Learning models** (e.g., Random Forest, LSTM) for improved forecast accuracy.
- **Multi-location Comparison**
    - Allow side-by-side visualization and comparison of forecasts for multiple locations.
- **Weather Parameter Expansion**
    - Extend forecasting to include **humidity**, **wind speed**, or **rainfall** if such data is available.
- **Mobile & Dashboard Version**
    - Convert the app into a **mobile-responsive dashboard** using shinydashboard or shinyMobile.
- **User Authentication and History**
    - Implement user login to save forecasts and track changes over time.
- **Export Feature**
    - Allow users to **download** forecast reports as PDF or Excel for offline analysis.

CHAPTER 5

APPENDIX A SOURCE CODE

```
library(shiny)
library(ggplot2)
library(zoo)
library(dplyr)
library(scales)

ui <- fluidPage(
  tags$head(
    tags$style(HTML("
      body {
        background-color: #f7f9fc;
      }
      .panel {
        background-color: #ffffff;
        border: 1px solid #ccc;
        border-radius: 10px;
        padding: 20px;
        margin-top: 10px;
      }
      h4 {
        color: #2c3e50;
      }
      .btn {
        background-color: #0073e6;
        color: white;
```

```
    }
    .btn:hover {
      background-color: #005bb5;
    }
  "))
 ),

 navbarPage("🌡️ Temperature Forecast App",

  tabPanel("🔧 Input Panel",
   sidebarLayout(
    sidebarPanel(
     fileInput("file", "Upload CSV (Location, Date, Temperature)", accept = ".csv"),
     uiOutput("locationSelect"),
     dateInput("from_date", "Start Forecast From:", value = Sys.Date(), format = "dd/mm/yyyy"),
     numericInput("forecast_days", "Days to Forecast", value = 7, min = 1),
     numericInput("sma_window", "SMA Window (days)", value = 3, min = 1),
     actionButton("analyze", "🔍 Generate Forecast", class = "btn")
    ),
    mainPanel(
     div(class = "panel",
       h4("📥 Please configure input and click Generate Forecast."),
       p("Once the forecast is generated, check the 'Forecast Output' tab.")
     )
    )
```

```
      )
    ),

    tabPanel("📊 Forecast Output",
      fluidRow(
        column(12, div(class = "panel", plotOutput("tempPlot")))
      ),
      fluidRow(
        column(12, div(class = "panel",
          h4("🗓️ Forecast Table"),
          tableOutput("forecastTable")
        ))
      ),
      fluidRow(
        column(12, div(class = "panel", uiOutput("forecastInsight")))
      )
    )
  )
)

server <- function(input, output, session) {
  # Load CSV
  raw_data <- reactive({
    req(input$file)
    df <- read.csv(input$file$datapath)
    colnames(df) <- tolower(colnames(df))

    df <- df %>%
```

```
    mutate(date = as.Date(date, tryFormats = c("%d/%m/%Y", "%Y-%m-%d",
"%m/%d/%Y")),
           location = as.factor(location)) %>%
    arrange(location, date)
  df
 })

 # Location selector
 output$locationSelect <- renderUI({
  req(raw_data())
  selectInput("selected_location", "Select Location", choices =
unique(raw_data()$location))
 })

 # Forecast
 forecast_data <- eventReactive(input$analyze, {
  req(raw_data(), input$selected_location, input$forecast_days,
input$from_date)
  df <- raw_data() %>% filter(location == input$selected_location)

  # Use data only before selected forecast date to train SMA
  train_data <- df %>% filter(date <= input$from_date)
  train_data$sma <- rollmean(train_data$temperature, input$sma_window, fill
= NA, align = "right")

  start_date <- input$from_date + 1
  forecast_dates <- seq(start_date, by = "day", length.out =
input$forecast_days)
```

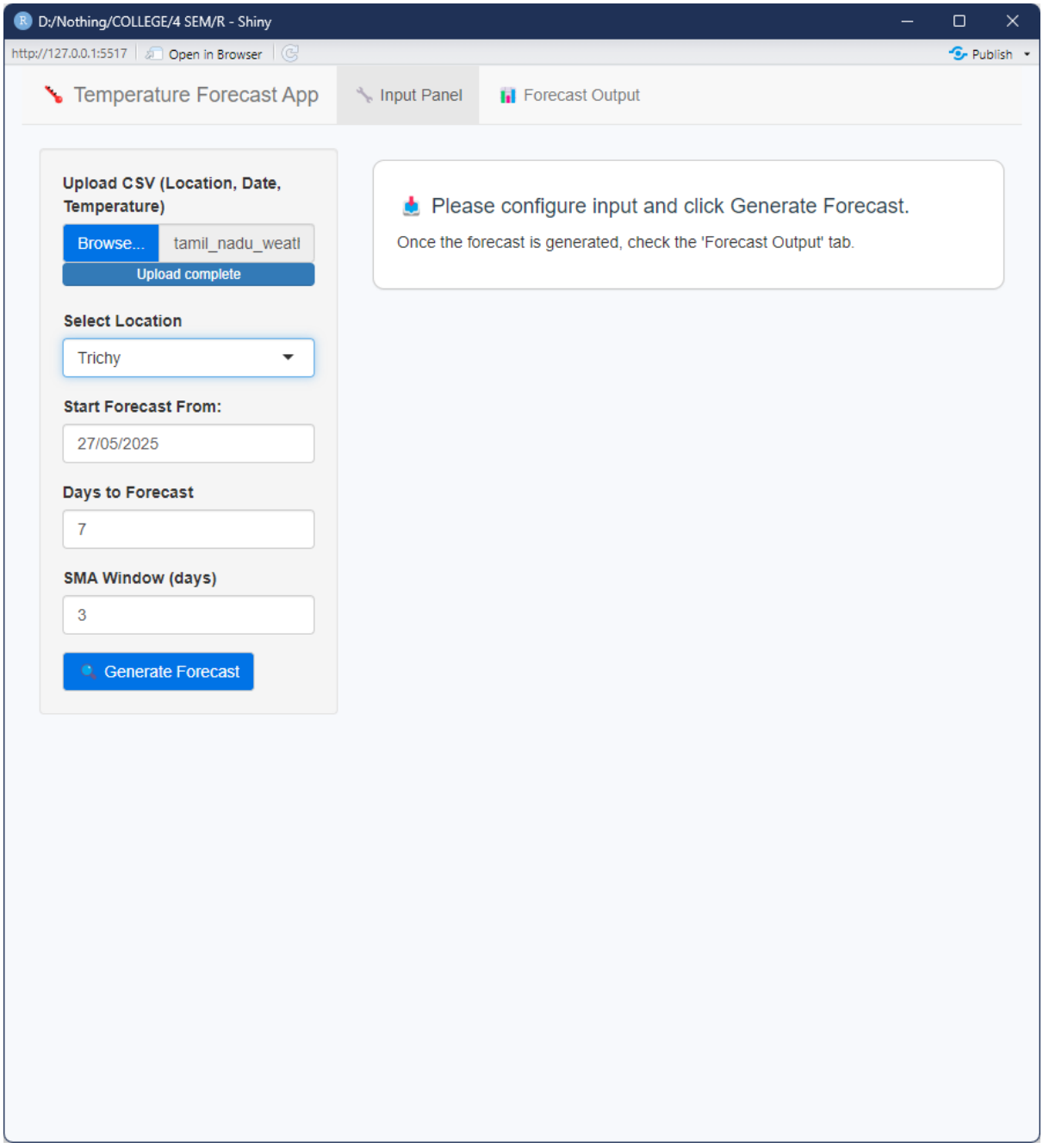
```
    forecast_temp <- rep(tail(train_data$sma[!is.na(train_data$sma)], 1),
input$forecast_days)

    forecast_df <- data.frame(
      location = input$selected_location,
      date = forecast_dates,
      temperature = forecast_temp,
      forecast = TRUE
    )

    historical_df <- df %>%
      filter(date <= input$from_date) %>%
      select(location, date, temperature) %>%
      mutate(forecast = FALSE)

    bind_rows(historical_df, forecast_df)
  })

  # Plot output
  output$tempPlot <- renderPlot({
    df <- forecast_data()
    ggplot(df, aes(x = date, y = temperature, color = forecast)) +
      geom_line(size = 1.2) +
      geom_point(size = 2) +
      scale_color_manual(values = c("FALSE" = "#34495e", "TRUE" =
"#e74c3c"), labels = c("Actual", "Forecast")) +
      labs(title = paste("Temperature Forecast -", input$selected_location),
           x = "Date", y = "Temperature (°C)", color = "Legend") +
      scale_x_date(labels = date_format("%d/%m/%Y")) +
```

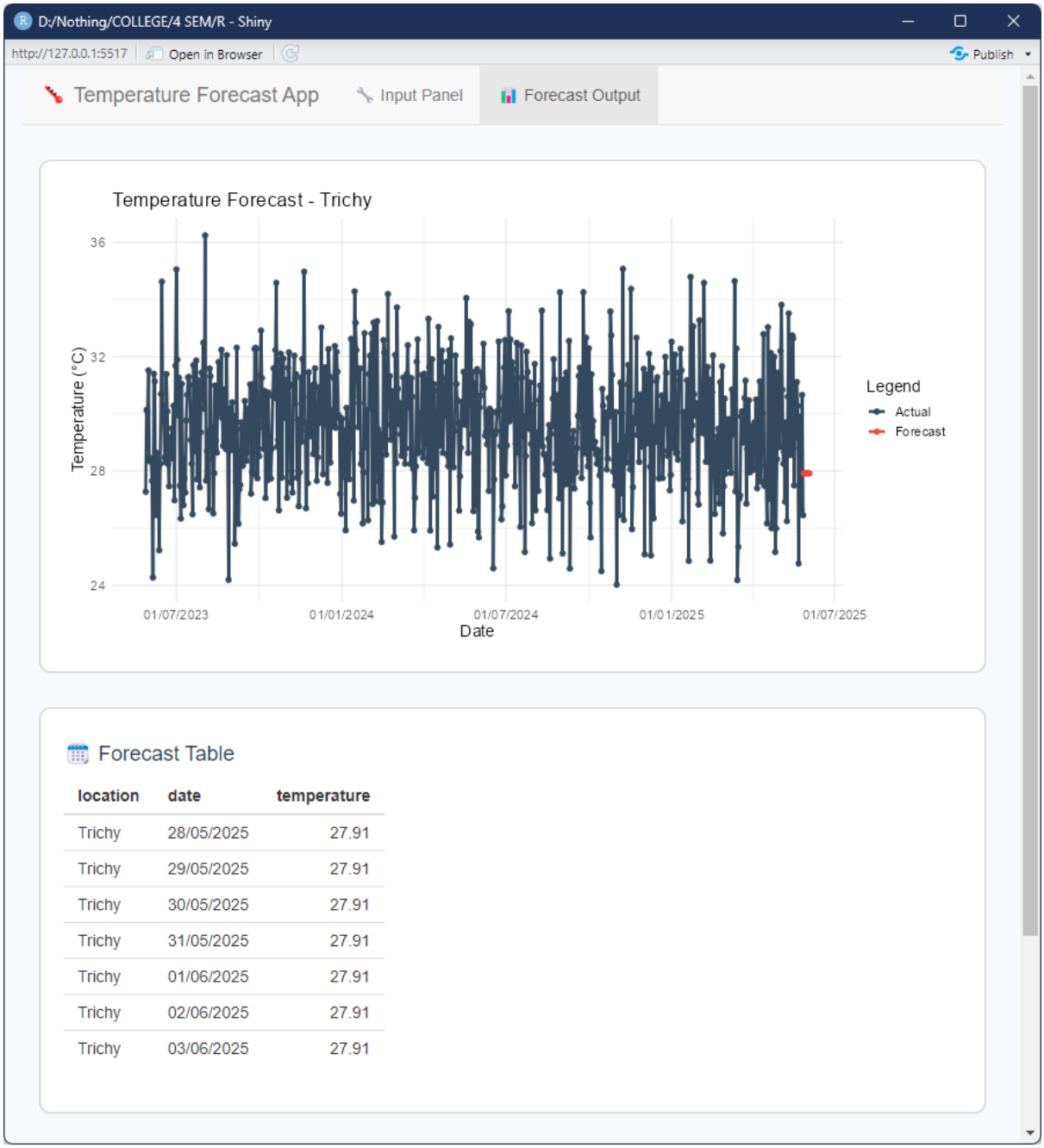
```
    theme_minimal(base_size = 14)
  })

  # Forecast table
  output$forecastTable <- renderTable({
    df <- forecast_data()
    df %>%
      filter(forecast == TRUE) %>%
      select(location, date, temperature) %>%
      mutate(date = format(date, "%d/%m/%Y"))
  })

  # Insight box
  output$forecastInsight <- renderUI({
    df <- forecast_data()
    forecast_df <- df %>% filter(forecast == TRUE)
    avg_forecast <- round(mean(forecast_df$temperature), 2)
    recent_avg <- round(mean(tail(df$temperature[df$forecast == FALSE],
input$sma_window)), 2)

    trend <- if (avg_forecast > recent_avg) "Rising ▲" else if (avg_forecast <
recent_avg) "Dropping ▼" else "Stable □"
    trend_color <- if (trend == "Rising ▲") "green" else if (trend == "Dropping
▼") "red" else "orange"

    HTML(paste0(
      "<h4>📊 Forecast Insight for <b>", input$selected_location, "</b></h4>",
```

```
      "<b>Forecast Start Date:</b> ", format(input$from_date + 1,
"%d/%m/%Y"), "<br>",
      "<b>Forecast Days:</b> ", input$forecast_days, "<br>",
      "<b>Recent ", input$sma_window, "-Day Avg Temp:</b> ", recent_avg,
"°C<br>",
      "<b>Forecasted Avg Temp:</b> ", avg_forecast, "°C<br>",
      "<b>Trend:</b> <span style='color:", trend_color, "; font-weight:bold;'>",
trend, "</span>"
    ))
  })
}

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

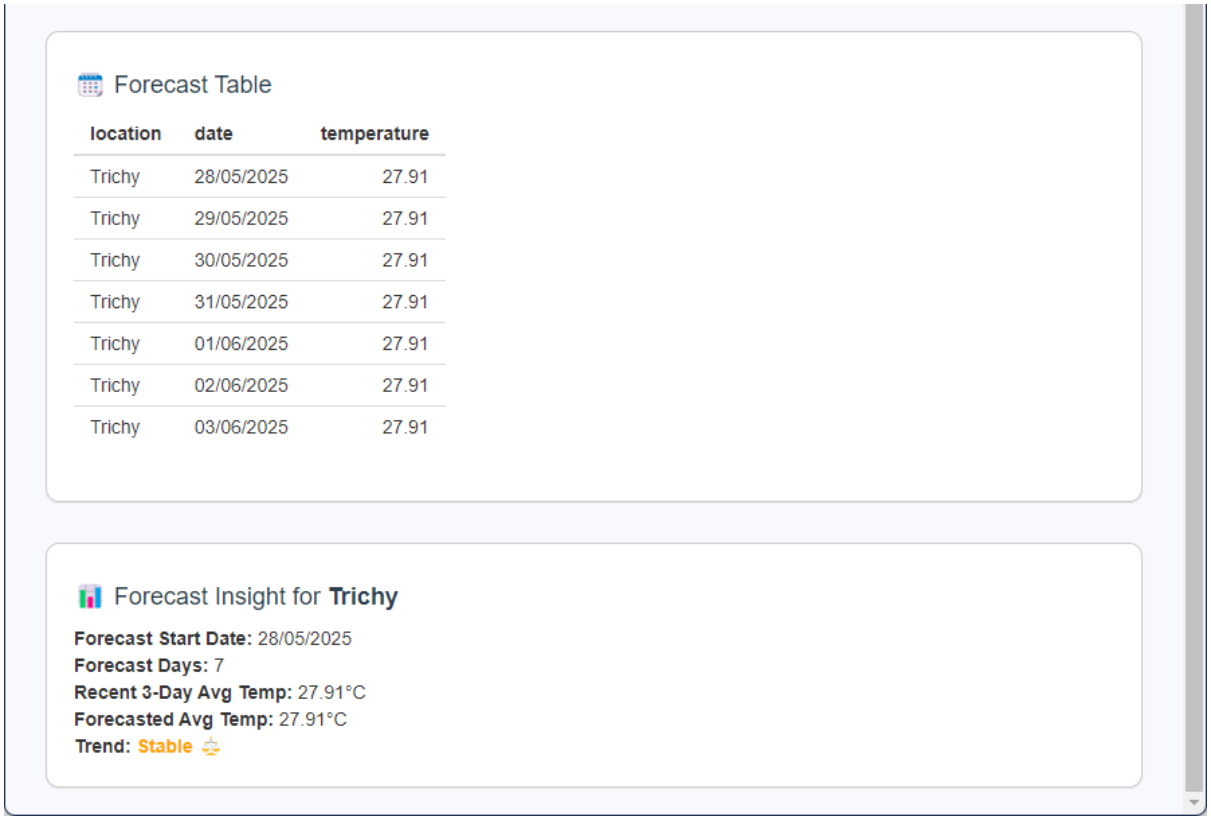
CHAPTER 6

APPENDIX B SCREENSHOTS





location	date	temperature
Trichy	28/05/2025	27.91
Trichy	29/05/2025	27.91
Trichy	30/05/2025	27.91
Trichy	31/05/2025	27.91
Trichy	01/06/2025	27.91
Trichy	02/06/2025	27.91
Trichy	03/06/2025	27.91


Forecast Table
location date temperature
Trichy 28/05/2025 27.91
Trichy 29/05/2025 27.91
Trichy 30/05/2025 27.91
Trichy 31/05/2025 27.91
Trichy 01/06/2025 27.91
Trichy 02/06/2025 27.91
Trichy 03/06/2025 27.91
Forecast Insight for Trichy
Forecast Start Date: 28/05/2025
Forecast Days: 7
Recent 3-Day Avg Temp: 27.91°C
Forecasted Avg Temp: 27.91°C
Trend: Stable

CHAPTER 7

REFERENCES

- Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2021). Shiny: Web Application Framework for R. R package version 1.7.1.
    - https://shiny.rstudio.com
- Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd ed.).
    - https://otexts.com/fpp2/
    - A comprehensive guide on forecasting techniques, including SMA and ARIMA models.
- R Core Team (2023). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
    - https://www.r-project.org
- Weather Data Simulation Inspired by Patterns from:
    - https://www.timeanddate.com/weather/india

- Wickham, H., & Grolemund, G. (2016). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media.
    - A foundational guide for data science using R, including packages like dplyr, ggplot2, and tidyr.