“World Clock Widget”

A Project Report Submitted in the partial fulfillment of the requirements of the course titled

“Problem Solving Through Programming (JAVA)”

BACHELOR OF TECHNOLOGY

In

DEPARTMENT OF FRESHMAN ENGINEERING

By

P.Pranaw reddy	**2520030136**
Aswika Inturi	**2520030221**
Varshini Bolla	**2520030431**
Marreddy Sai Bhavitha reddy	**2520030551**
Hamsini Vanam	**2520030556**
Vedadri naidu Pinninti	**2520090060**

Under the Esteemed Guidance of

Dr Jayaram Boga
Assistant professor
Department of Freshman Engineering

(DEEMED TO BE UNIVERSITY)

Koneru Lakshmaiah Education Foundation
(Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Off-Campus: Bachupally-Gandimaisamma Road, Bowrampet, Hyderabad, Telangana - 500 043.
Phone No: 7815926816, www.klh.edu.in

K L (Deemed to be) University

DEPARTMENT OF FRESHMAN ENGINEERING

Declaration

The Project Report entitled " World Clock Widget " is a record of Bonafide work of P.Pranaw reddy-2520030136, Aswika Inturi-2520030221, **Varshini Bolla-**2520030431, **Marreddy Sai Bhavitha reddy** -2520030551, **Hamsini Vanam** -2520030556, **Vedadri naidu Pinninti** -2520090060 **submitted in partial fulfillment of the requirements of the course titled "Problem Solving Through Programming (JAVA)" under the B.Tech Ist Year Trimester - I program in Department of Freshman Engineering at K L University. The results presented in this report have not been copied from any other department, university, or institute.**

P.Pranaw reddy-2520030136

Aswika Inturi-2520030221

Varshini Bolla-2520030431

Marreddy Sai Bhavitha reddy -2520030551

Hamsini Vanam -2520030556

Vedadri naidu Pinninti - 2520090060

K L (Deemed to be) University

DEPARTMENT OF FRESHMAN ENGINEERING

CERTIFICATE

This is certify that the java project based report entitled **"World Clock Widget"** is a bonafide work done and submitted P.Pranaw reddy-2520030136, **Aswika Inturi-2520030221, Varshini Bolla-2520030431, Marreddy Sai Bhavitha reddy -2520030551, Hamsini Vanam -2520030556, Vedadri naidu Pinninti -2520090060** in partial fulfillment of the requirements of the course titled "Problem Solving Through Programming (JAVA)" under the B.Tech Ist Year Trimester - I program in Department of Freshman Engineering, K L (Deemed to be University), during the academic year **2025-2026.**

Signature of the Guide

Signature of the Course Coordinator

Signature of the HOD

ACKNOWLEDGEMENT

The success in this project would not have been possible but for the timely help and guidance rendered by many people. Our wish to express my sincere thanks to all those who has assisted us in one way or the other for the completion of my project.

Our greatest appreciation to my Course Coordinator **Dr Y Ashok**, and my guide **Dr Jayaram Boga ,** Department of Freshman Engineering which cannot be expressed in words for his/her tremendous support, encouragement and guidance for this project.

We express our gratitude to **Dr. N. Chaitanya Kumar**, Head of the ***Department for Freshman Engineering*** for providing us with adequate facilities, ways and means by which we are able to complete this project-based Lab.

We thank all the members of teaching and non-teaching staff members, and also who have assisted me directly or indirectly for successful completion of this project. Finally, I sincerely thank my parents, friends and classmates for their kind help and cooperation during my work.

P.Pranaw reddy-2520030136

Aswika Inturi-2520030221

Varshini Bolla-2520030431

Marreddy Sai Bhavitha reddy -2520030551

Hamsini Vanam -2520030556
Vedadri naidu Pinninti - 2520090060

Abstract

This project, *World Clock Widget*, is a lightweight, draggable Java Swing application designed to display accurate global time across multiple countries. The widget provides real-time updates of the selected country's date and time while simultaneously displaying converted times of several other international time zones. It offers a clean, minimalistic design and supports both dark and light themes, making it suitable for desktop use.

The project incorporates several Java concepts, including GUI development using Swing, event handling, time zone conversion using ZonedDateTime, user interaction support through mouse listeners, and system tray integration. The application enables users to minimize the clock as a tray icon, restore it on demand, and drag the widget anywhere on the desktop.

This project demonstrates strong understanding of Java programming fundamentals, object-oriented techniques, graphical components, and real-world application development skills. It serves as a practical example of problem-solving through programming and showcases the ability to design a functional, user-friendly desktop utility.

Index

CHAPTER -1 INTRODUCTION

1.1Background of the Project

In today's globalized world, people frequently interact with colleagues, friends, businesses, and services across different time zones. Knowing the accurate local time of multiple regions is essential for communication, travel planning, and international coordination. Traditional system clocks show only the local time, and switching between time zones can be inconvenient.

To solve this issue, the *World Clock Widget* was developed as a small, always-on-top desktop tool. The widget remains visible on the user's screen and displays real-time updates of selected global regions. It reduces the need for manual conversions or checking external websites. The application is intentionally lightweight, minimal, and user-friendly, ensuring functionality without occupying significant screen space.

This project was chosen because it combines Java Swing GUI development with real-world utility, enhancing practical knowledge and fostering confidence in building interactive desktop applications.

1.2 Problem Statement

Users often face difficulty in keeping track of time in different countries, especially while working or communicating internationally. Most operating systems display only local time, requiring users to open browsers or external applications to check global times. This leads to inconvenience and loss of productivity.

The problem can be summarized as:

- Lack of easy access to multiple global time zones
- No simple draggable widget for desktop
- Need for real-time conversion and synchronization
- Requirement for a minimal and always-on-top clock utility

The proposed solution is a Java-based real-time *World Clock Widget* that solves these issues effectively.

CHAPTER -2 SYSTEM ARCHITECTURE

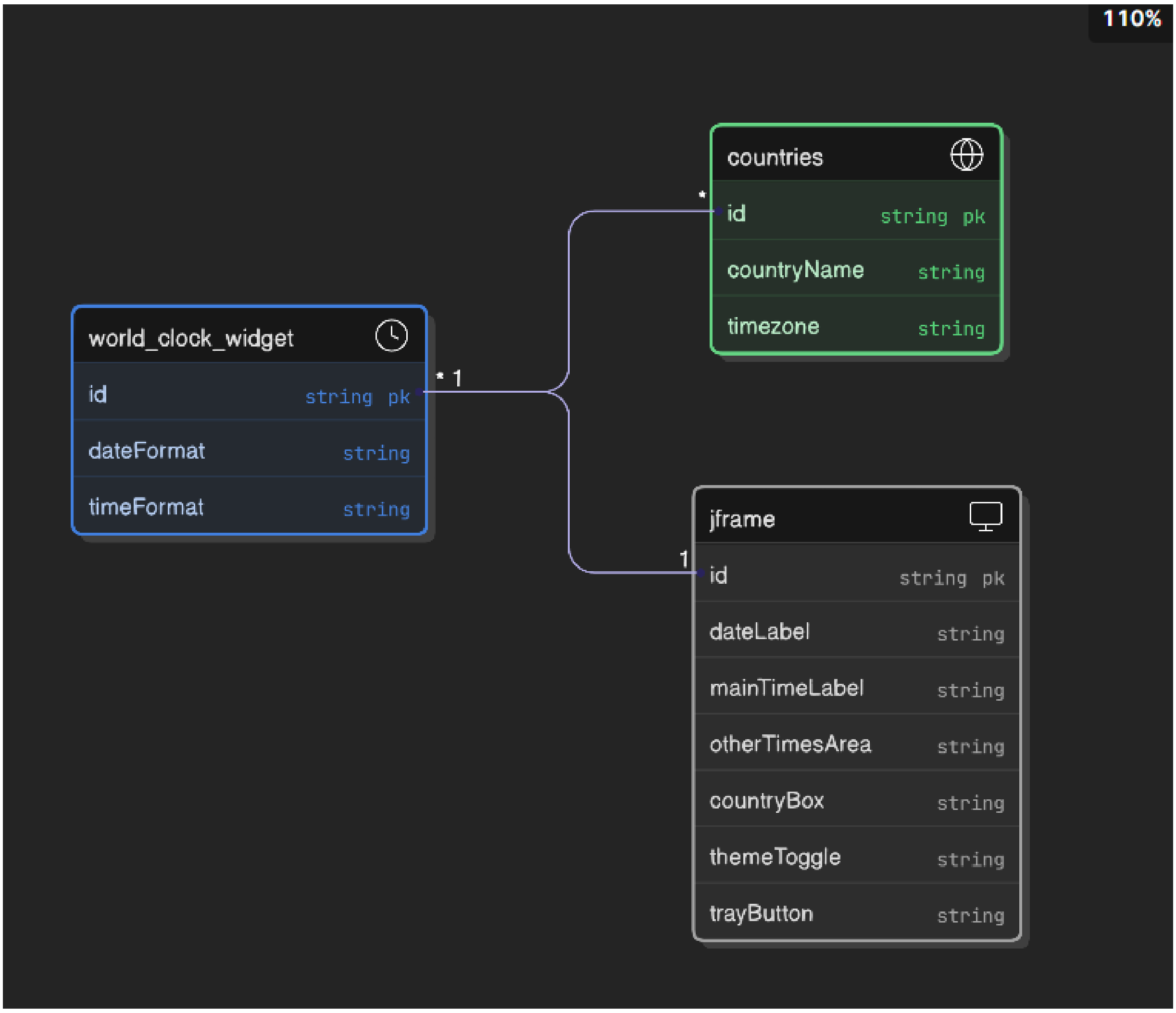
2.1High-level architecture diagram

The system architecture of the World Clock Widget consists of:

1. User Interface Layer (Swing Components)
    - JFrame
    - JLabels
    - JTextArea
    - JComboBox
    - Buttons and Toggle buttons
2. Logic Layer
    - Time zone calculation
    - Date/time formatting
    - Dynamic theme management
    - System tray operations
3. Event Handling Layer
    - Action listeners
    - Timer listeners
    - Mouse listeners for dragging
4. Data Layer
    - Country–TimeZone map
    - Conversions for all selected time zones

This layered architecture ensures modularity, maintainability, and easy future enhancements.

2.2 Class Diagram



world_clock_widget	
id	string pk
dateFormat	string
timeFormat	string

countries	
id	string pk
countryName	string
timezone	string

jframe	
id	string pk
dateLabel	string
mainTimeLabel	string
otherTimesArea	string
countryBox	string
themeToggle	string
trayButton	string

CHAPTER -3 CO's ATTAINMENT

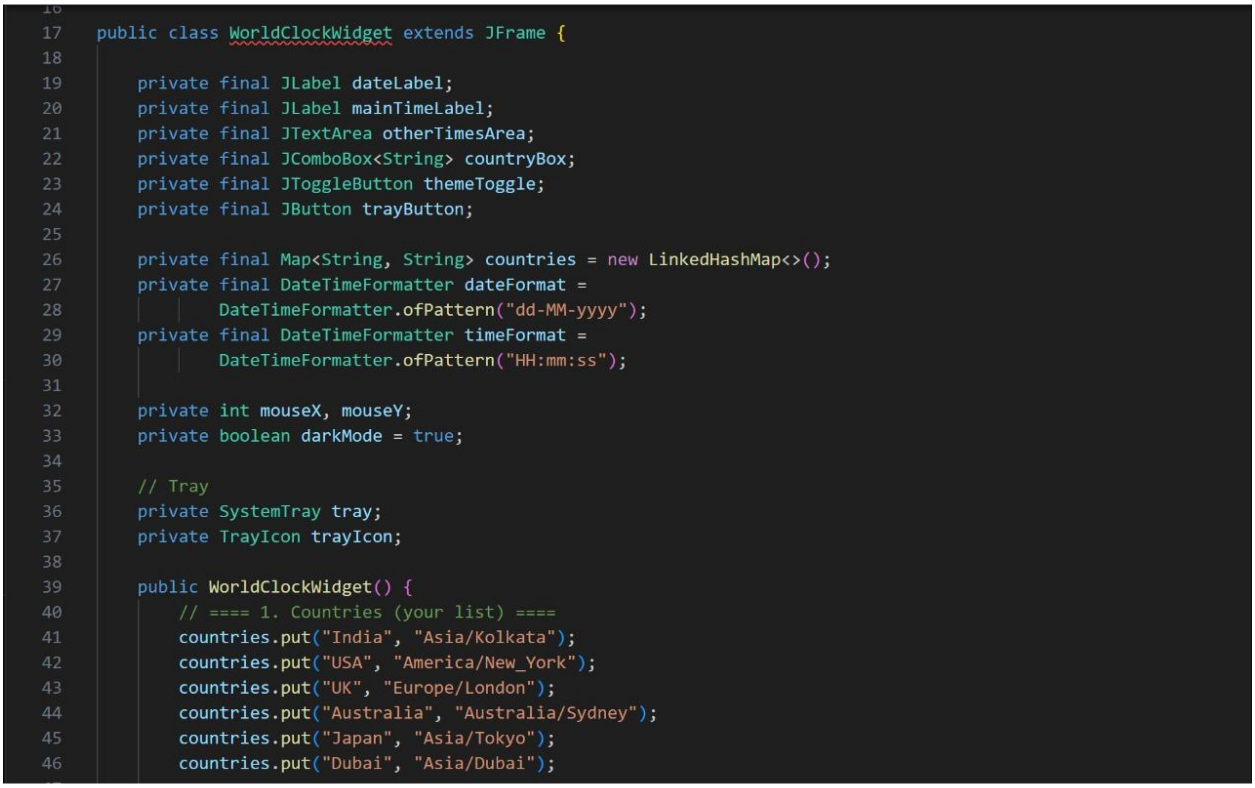
3.1 CO1 Attainment

CO1 Syllabus	**CO1 Concepts Included in Project**
Basic Java programming constructs including variables, data types, operators, loops, conditionals, and methods.	• Variables and data types for storing time zones • Looping to iterate over countries and generate converted times • Conditional logic for theme switching • Methods used for modular code structure

3.1.1 Scenario's for CO1 implementation.

The updateTimes() method uses loops and string formatting to dynamically update global times every second.

3.1.2 CO1 code screen shot.



```java
public class WorldClockWidget extends JFrame {

    private final JLabel dateLabel;
    private final JLabel mainTimeLabel;
    private final JTextArea otherTimesArea;
    private final JComboBox<String> countryBox;
    private final JToggleButton themeToggle;
    private final JButton trayButton;

    private final Map<String, String> countries = new LinkedHashMap<>();
    private final DateTimeFormatter dateFormat =
            DateTimeFormatter.ofPattern("dd-MM-yyyy");
    private final DateTimeFormatter timeFormat =
            DateTimeFormatter.ofPattern("HH:mm:ss");

    private int mouseX, mouseY;
    private boolean darkMode = true;

    // Tray
    private SystemTray tray;
    private TrayIcon trayIcon;

    public WorldClockWidget() {
        // ==== 1. Countries (your list) ====
        countries.put("India", "Asia/Kolkata");
        countries.put("USA", "America/New_York");
        countries.put("UK", "Europe/London");
        countries.put("Australia", "Australia/Sydney");
        countries.put("Japan", "Asia/Tokyo");
        countries.put("Dubai", "Asia/Dubai");
```

3.2 CO2 Attainment: Implement Object-Oriented Programming

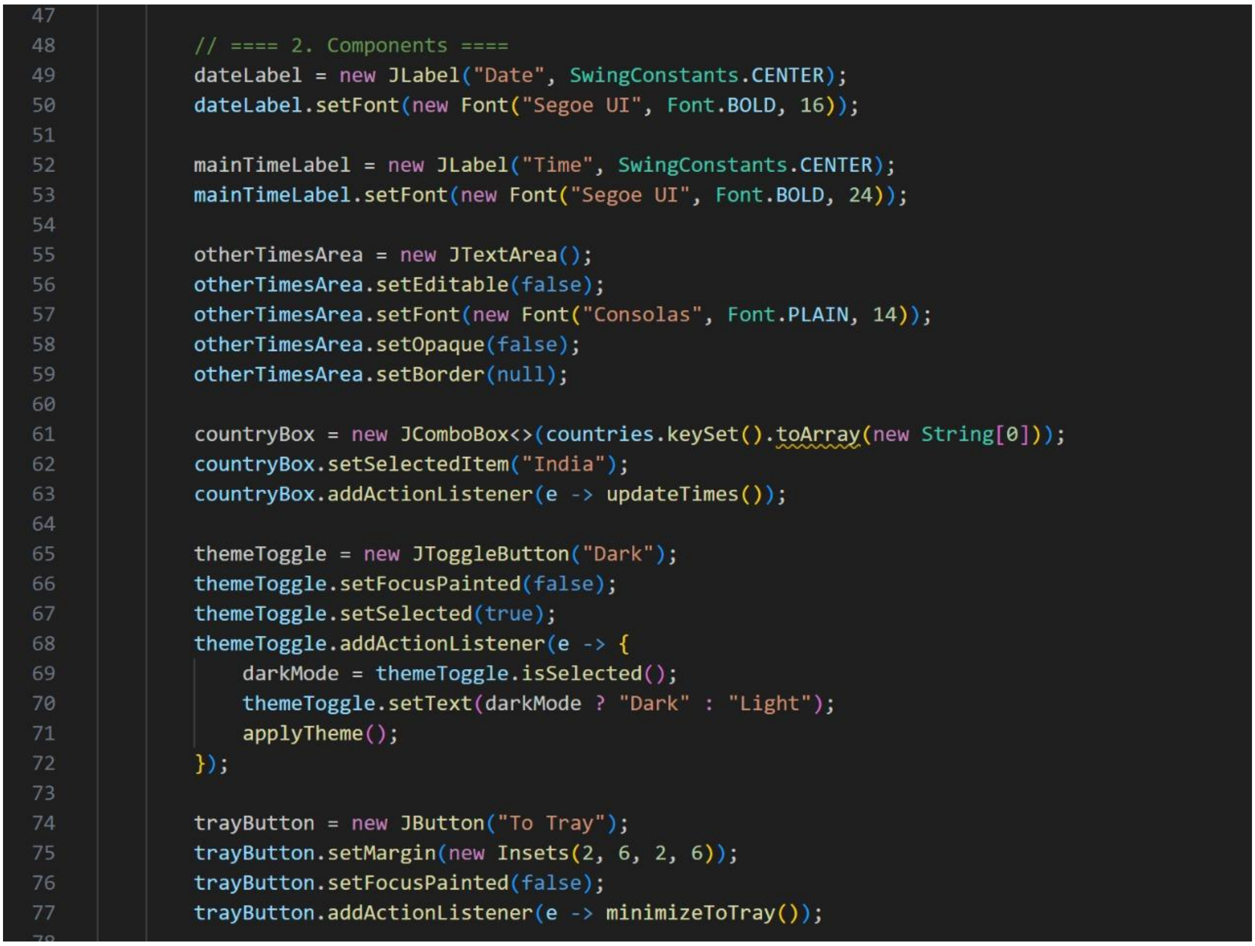
Concepts Used

- Encapsulation
- Classes & objects
- Managing UI elements as private attributes
- Constructor for initializing interface components

Scenario

The entire widget is implemented as a single class with multiple methods, each handling specific functionality.

3.2.2 CO2 code screen shot.



```

        // ==== 2. Components ====
        dateLabel = new JLabel("Date", SwingConstants.CENTER);
        dateLabel.setFont(new Font("Segoe UI", Font.BOLD, 16));

        mainTimeLabel = new JLabel("Time", SwingConstants.CENTER);
        mainTimeLabel.setFont(new Font("Segoe UI", Font.BOLD, 24));

        otherTimesArea = new JTextArea();
        otherTimesArea.setEditable(false);
        otherTimesArea.setFont(new Font("Consolas", Font.PLAIN, 14));
        otherTimesArea.setOpaque(false);
        otherTimesArea.setBorder(null);

        countryBox = new JComboBox<>(countries.keySet().toArray(new String[0]));
        countryBox.setSelectedItem("India");
        countryBox.addActionListener(e -> updateTimes());

        themeToggle = new JToggleButton("Dark");
        themeToggle.setFocusPainted(false);
        themeToggle.setSelected(true);
        themeToggle.addActionListener(e -> {
            darkMode = themeToggle.isSelected();
            themeToggle.setText(darkMode ? "Dark" : "Light");
            applyTheme();
        });

        trayButton = new JButton("To Tray");
        trayButton.setMargin(new Insets(2, 6, 2, 6));
        trayButton.setFocusPainted(false);
        trayButton.addActionListener(e -> minimizeToTray());
```

3.3 CO3 Attainment: Develop Applications using GUI-based components

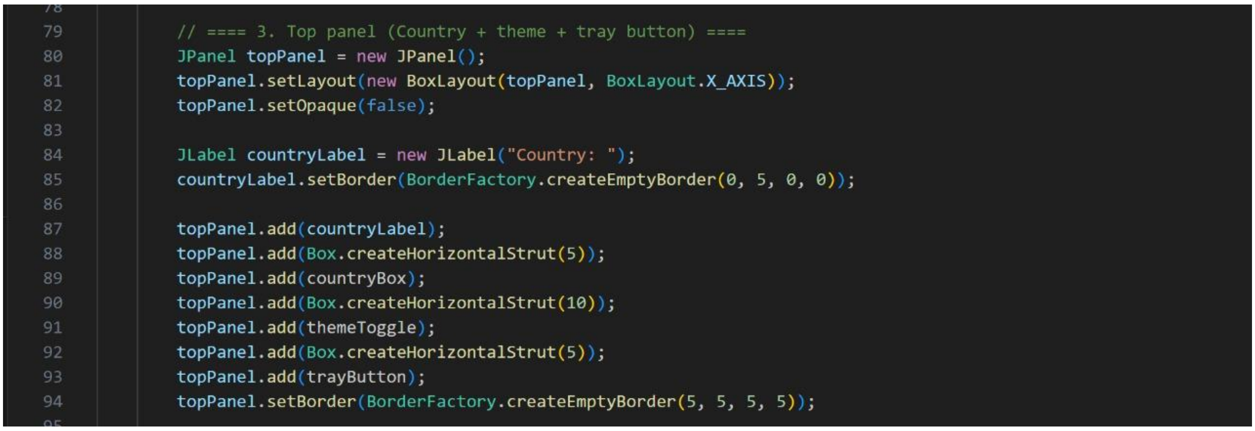
1. **Concepts Used**

- Swing components such as JFrame, JLabel, JComboBox, JTextArea

- Layout management (BorderLayout, GridLayout, BoxLayout)

- Handling GUI events

3.3.2 CO3 code screen shot.

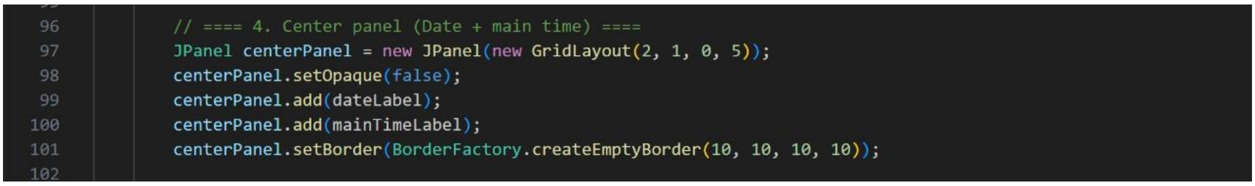


```
        // ==== 3. Top panel (Country + theme + tray button) ====
        JPanel topPanel = new JPanel();
        topPanel.setLayout(new BoxLayout(topPanel, BoxLayout.X_AXIS));
        topPanel.setOpaque(false);

        JLabel countryLabel = new JLabel("Country: ");
        countryLabel.setBorder(BorderFactory.createEmptyBorder(0, 5, 0, 0));

        topPanel.add(countryLabel);
        topPanel.add(Box.createHorizontalStrut(5));
        topPanel.add(countryBox);
        topPanel.add(Box.createHorizontalStrut(10));
        topPanel.add(themeToggle);
        topPanel.add(Box.createHorizontalStrut(5));
        topPanel.add(trayButton);
        topPanel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));
```

3.4 CO4 Attainment: Implement event-driven programming
Concepts Used

- ActionListener for buttons
- MouseAdapter for dragging
- Timer for continuous time updates
- Tray actions for window restoration

3.4.2 CO4 code screen shot.

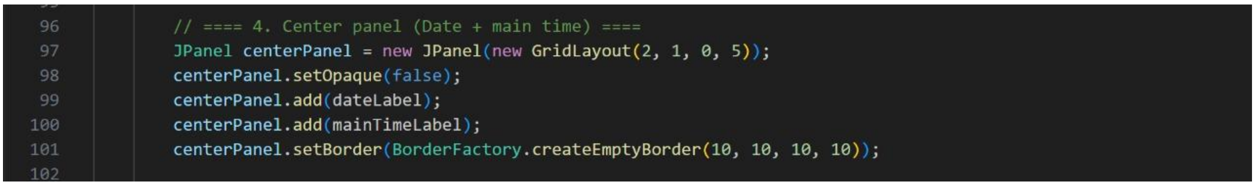


```
        // ==== 4. Center panel (Date + main time) ====
        JPanel centerPanel = new JPanel(new GridLayout(2, 1, 0, 5));
        centerPanel.setOpaque(false);
        centerPanel.add(dateLabel);
        centerPanel.add(mainTimeLabel);
        centerPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
```

3.5 CO5 Attainment: Apply problem-solving and design thinking Concepts Used

- Designing a minimal, user-friendly tool
- Creating light/dark theme logic
- Combining multiple features (draggable UI, system tray, real-time clock)

3.5.2 CO5 code screen shot.

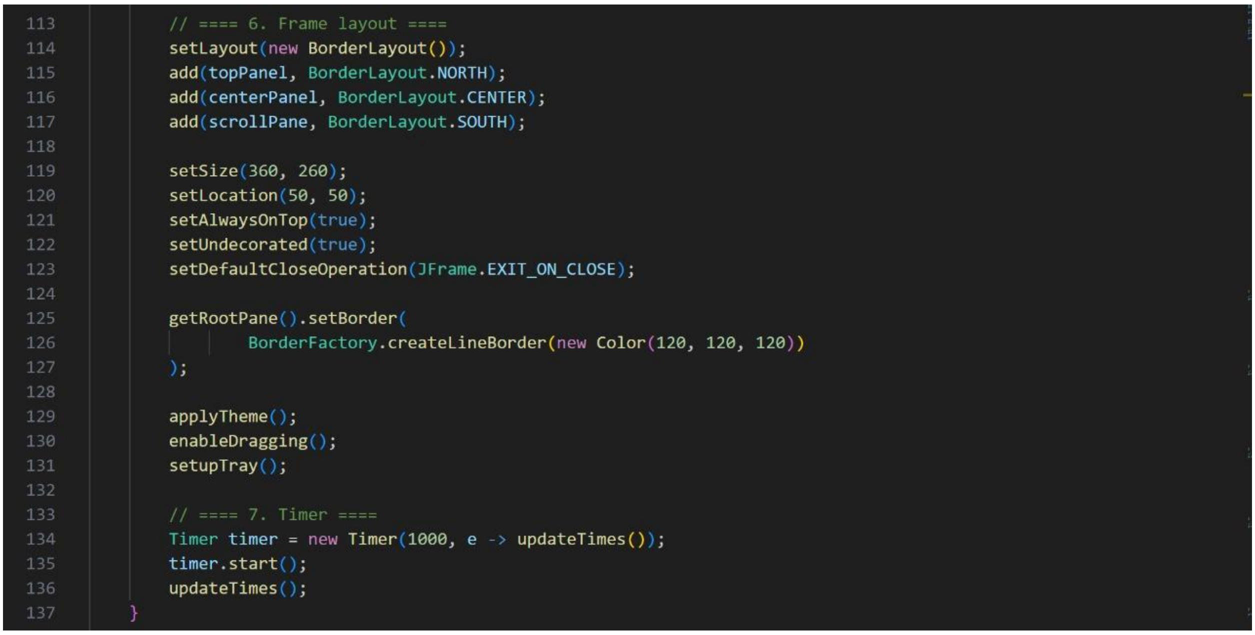


```
        // ==== 4. Center panel (Date + main time) ====
        JPanel centerPanel = new JPanel(new GridLayout(2, 1, 0, 5));
        centerPanel.setOpaque(false);
        centerPanel.add(dateLabel);
        centerPanel.add(mainTimeLabel);
        centerPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
```

3.6 CO6 Attainment: Execute a working project with documentation Concepts Used

- Complete development lifecycle
- Documentation, screenshots, testing, enhancements

3.6.2 CO6 code screen shot.

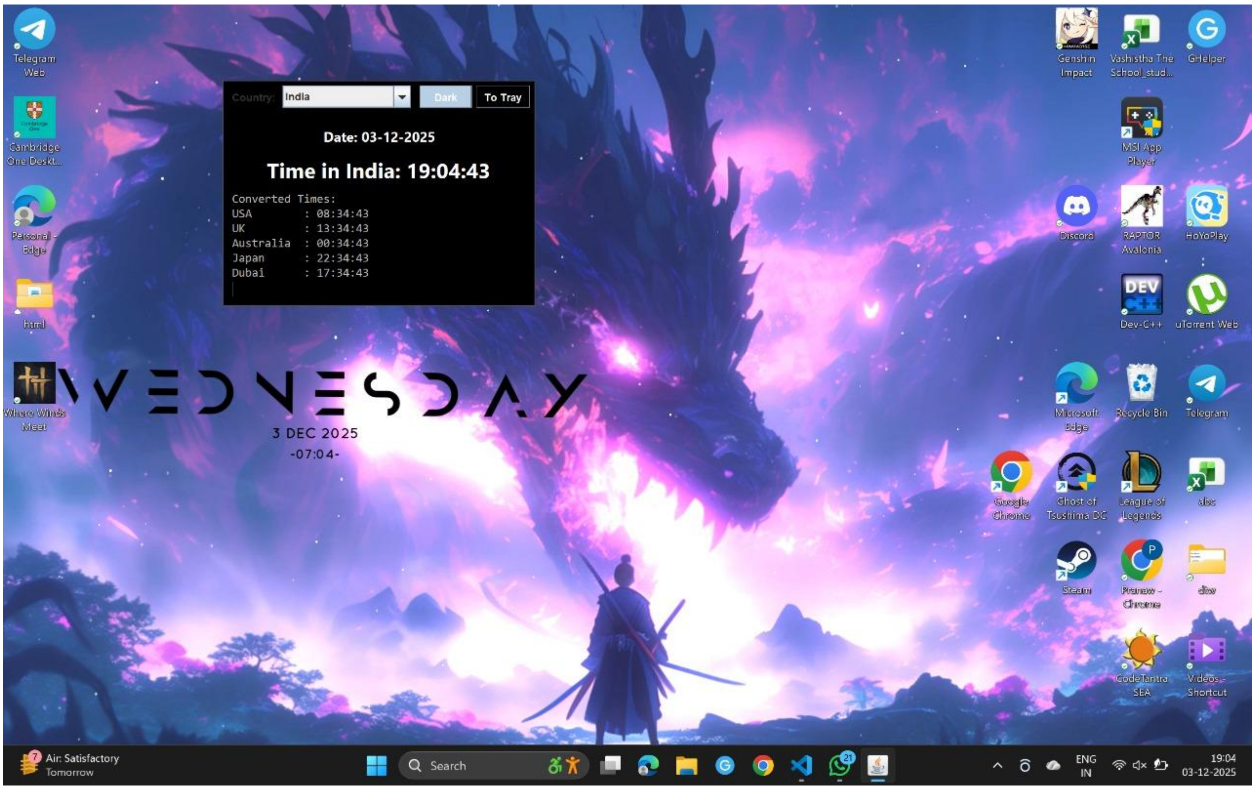


```
        // ==== 6. Frame layout ====
        setLayout(new BorderLayout());
        add(topPanel, BorderLayout.NORTH);
        add(centerPanel, BorderLayout.CENTER);
        add(scrollPane, BorderLayout.SOUTH);

        setSize(360, 260);
        setLocation(50, 50);
        setAlwaysOnTop(true);
        setUndecorated(true);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        getRootPane().setBorder(
                BorderFactory.createLineBorder(new Color(120, 120, 120))
        );

        applyTheme();
        enableDragging();
        setupTray();

        // ==== 7. Timer ====
        Timer timer = new Timer(1000, e -> updateTimes());
        timer.start();
        updateTimes();
    }
```

CHAPTER -4 SCREEN SHOTS

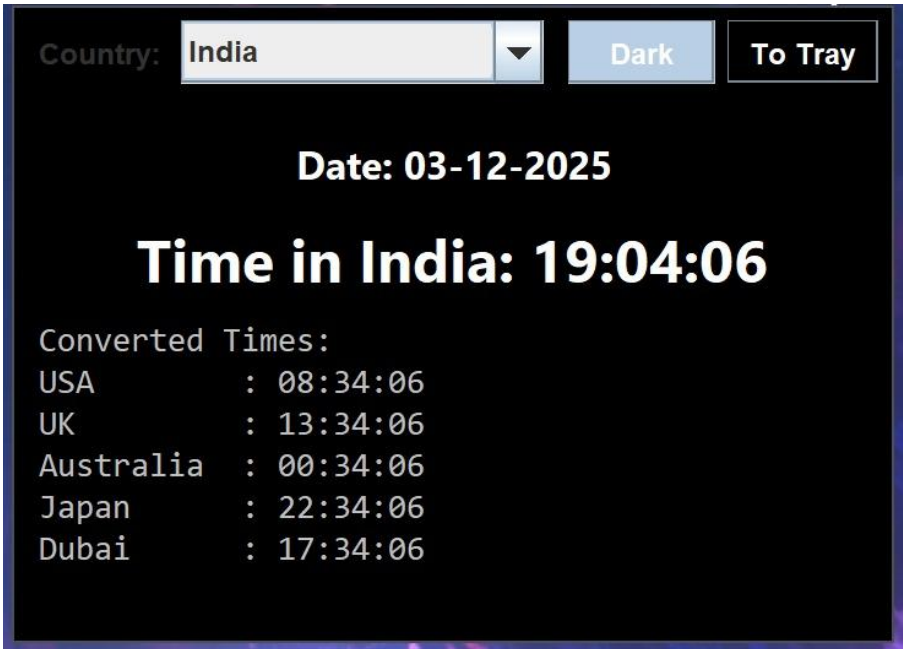
4.1 Screen Shots

1.Execution



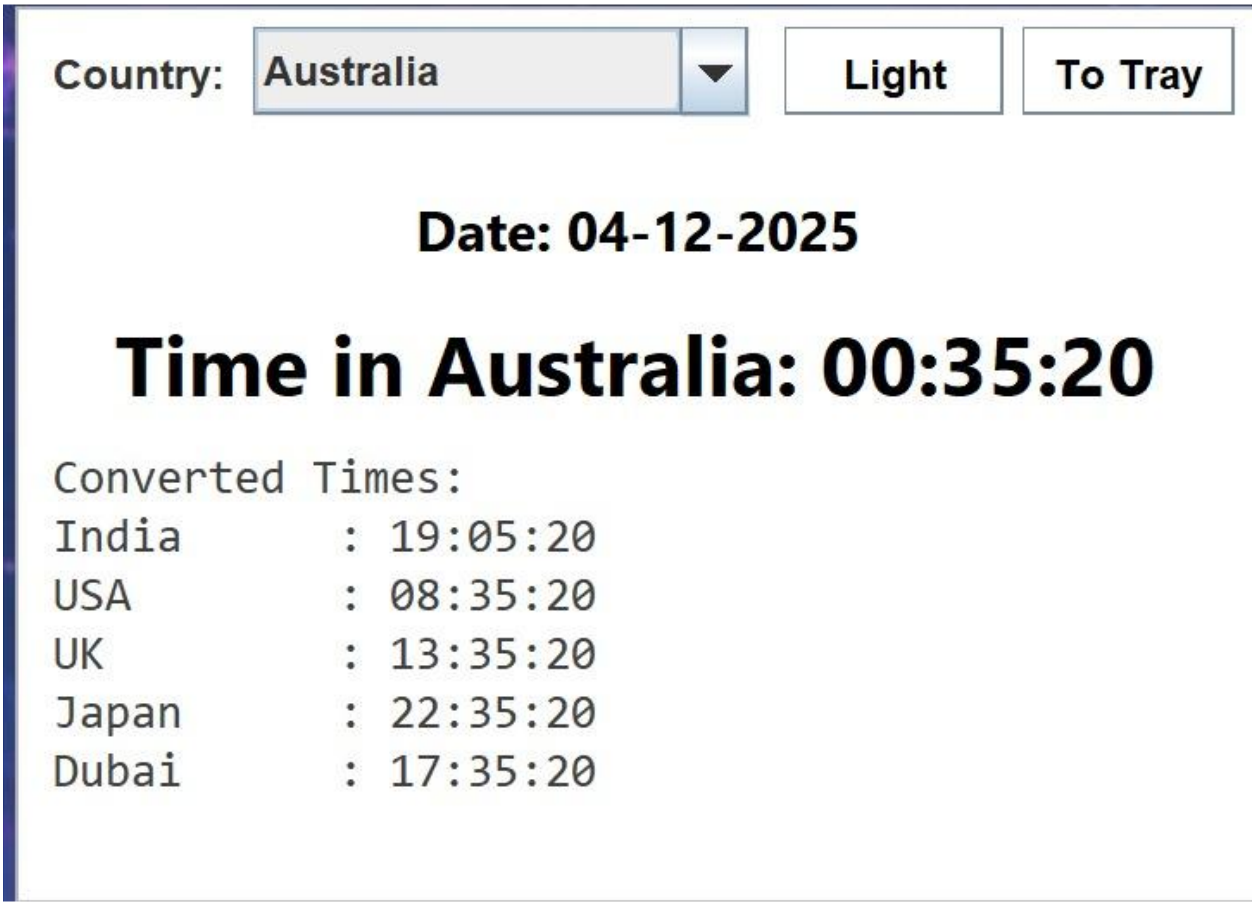
```
Date: 03-12-2025

Time in India: 19:04:43

Converted Times:
USA        : 08:34:43
UK         : 13:34:43
Australia  : 00:34:43
Japan      : 22:34:43
Dubai      : 17:34:43
```

2.widget



```
Date: 03-12-2025

Time in India: 19:04:06

Converted Times:
USA        : 08:34:06
UK         : 13:34:06
Australia  : 00:34:06
Japan      : 22:34:06
Dubai      : 17:34:06
```

3.Light Mode

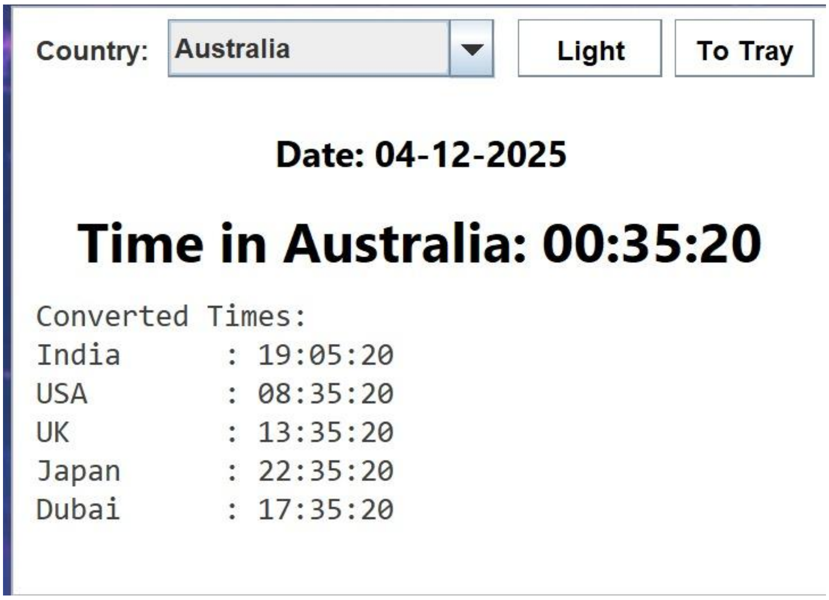


Date: 04-12-2025

Time in Australia: 00:35:20

```
Converted Times:
India        : 19:05:20
USA          : 08:35:20
UK           : 13:35:20
Japan        : 22:35:20
Dubai        : 17:35:20
```

4.Time in Australia

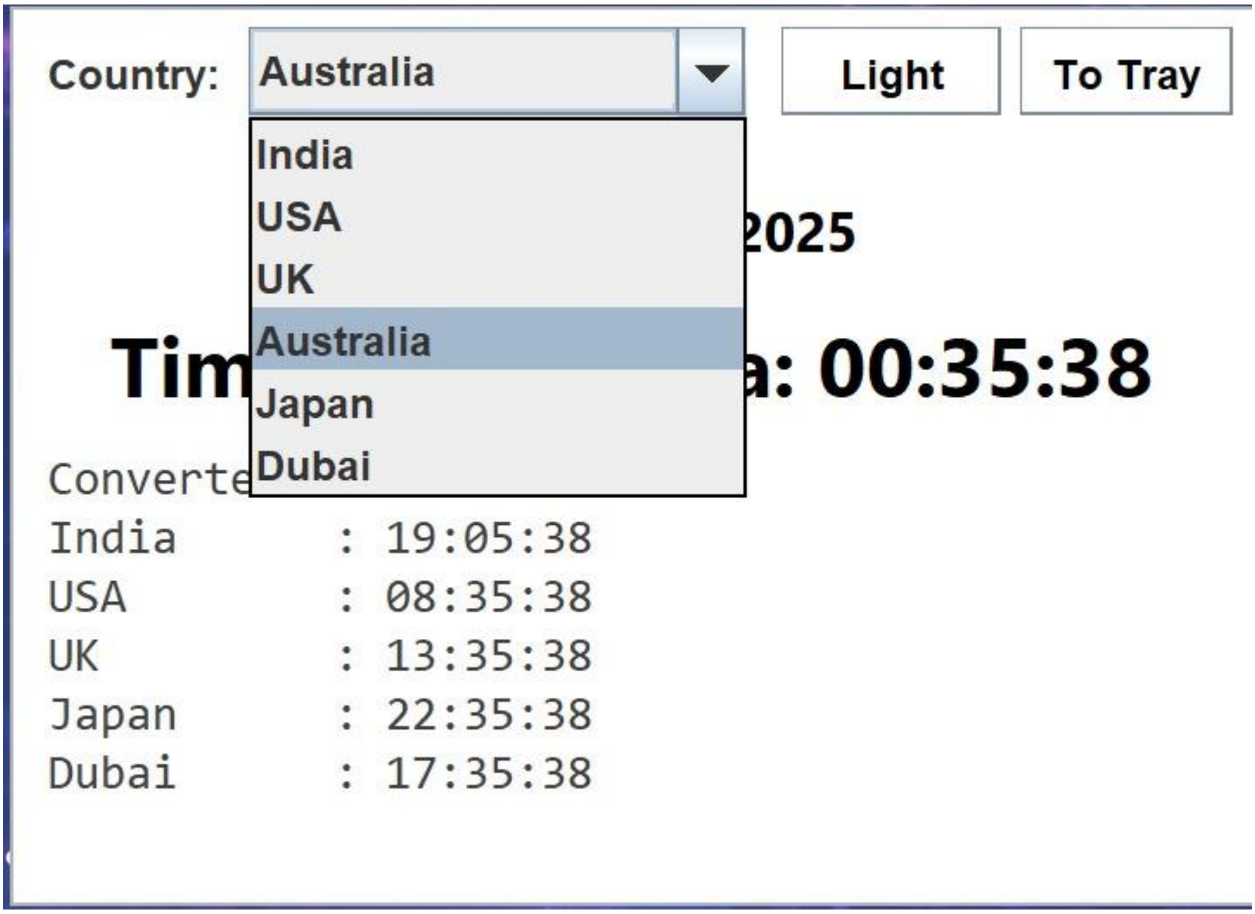


Date: 04-12-2025

Time in Australia: 00:35:20

```
Converted Times:
India        : 19:05:20
USA          : 08:35:20
UK           : 13:35:20
Japan        : 22:35:20
Dubai        : 17:35:20
```

5. Country options



Country: Australia

```
2025

Tim                    a: 00:35:38

Converte
India        : 19:05:38
USA          : 08:35:38
UK           : 13:35:38
Japan        : 22:35:38
Dubai        : 17:35:38
```

CHAPTER -5 TESTING

5.1Test Cases and Results

Test Case	Input	Expected Output	Result
Select country	USA	Correct USA time shown	Pass
Switch theme	Toggle	Background & text color change	Pass
Drag widget	Hold mouse	Widget moves smoothly	Pass
Minimize to tray	Press button	Icon appears in tray	Pass
Restore from tray	Click tray icon	Window reappears	Pass

CHAPTER -6 FUTURE ENHANCEMENTS

6.1Planned Features

1. Add analog clock rendering
2. Add more time zones
3. Provide alarms & reminders
4. Add custom themes
5. Add widget resizing options
6. Export selected time zones as PDF
7. Allow user to add/remove countries

CHAPTER -7 CONCLUSION

7.1Summary of the Project

The World Clock Widget project successfully demonstrates the use of Java programming skills to develop a real-world application. It combines GUI development, event-driven programming, time zone conversion, and user interaction through system tray and drag features. The project not only meets the functional requirements but also delivers a practical desktop tool that users can rely on.

Through this project, valuable skills such as designing interfaces, debugging, code optimization, and documentation were learned. Overall, this project strengthened the understanding of Java Swing, object-oriented principles, and software design techniques.

CHAPTER -8 REFERENCES

Books

- Schildt, H. (2018). *Java: The Complete Reference* (11th ed.). McGraw-Hill Education.
- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program* (10th ed.). Pearson.
- Horton, I. (2018). *Beginning Java SE 9* (8th ed.). Apress.
- Robillard, M. P. (2020). *Introduction to Software Design with Java*. Springer.
- Bishop, J. (2012). *Java Gently: Programming Principles Explained* (4th ed.). Addison-Wesley.

Tutorials and Learning Resources

- Oracle. (2025). *Java Swing Tutorial*. Retrieved from https://docs.oracle.com/javase/tutorial/uiswing
- Oracle. (2025). *Java Time API Documentation*. Retrieved from https://docs.oracle.com/javase/8/docs/api/java/time
- GeeksforGeeks. (2025). *Java Swing Tutorials and Examples*. Retrieved from https://www.geeksforgeeks.org/java-swing
- W3Schools. (2025). *Java Programming Basics*. Retrieved from https://www.w3schools.com/java
- Tutorialspoint. (2025). *Java Swing, AWT, and Event Handling*. Retrieved from https://www.tutorialspoint.com/java

APIs and Tools Used

- Oracle Java Development Kit (JDK). (2025). Retrieved from https://www.oracle.com/java/technologies
- IntelliJ IDEA / VS Code Java Extension Pack. (2025). Retrieved from https://code.visualstudio.com

- java.time API for time zone conversion (ZonedDateTime, ZoneId).
- Swing Framework for GUI components (JFrame, JLabel, JTextArea, JComboBox).
- SystemTray API (TrayIcon, PopupMenu) for desktop tray integration.

Documentation and Research References

- Oracle Docs. (2025). *SystemTray and Desktop Integration in Java*. Retrieved from https://docs.oracle.com/javase
- Stack Overflow. (2024–2025). Solutions and discussions related to Java Swing event handling and GUI design.
- Java Platform SE Documentation. (2025). *Event Handling, Listeners, and Swing Components*. Retrieved from https://docs.oracle.com/javase
- GitHub Repositories and Open-Source Java Swing Projects for design patterns and reference implementations.
- Baeldung. (2025). *Guide to Java Date and Time API*. Retrieved from https://www.baeldung.com/java-date-time

CHAPTER -9 APPENDICES

Appendix A: Installation / Setup Instructions

- Install JDK 8 or above
- Install VS Code or IntelliJ IDEA
- Add Java Extension Pack (if in VS Code)
- Compile the project using:

  ```
  javac WorldClockWidget.java
  ```
- Run the project using:

  ```
  java WorldClockWidget
  ```

Appendix B: User Manual

1. Select a country from the dropdown to view its time
2. Use the theme toggle to switch between Dark and Light modes
3. Click "To Tray" to minimize the widget to the system tray
4. Click and drag the clock to move it anywhere on the screen
5. Click the tray icon to restore the widget

Geo Tag photos with guide