Project report

WORLD CLOCK

WORLD CLOCK SYSTEM USING PYTHON :-

NAME:- YATHARTH SHUKLA

REG NO:-25BAI10751

CSE-AIML

ABSTRACT:-

The World Clock System is a Python-based application that shows the current time and date in multiple cities worldwide, accounting for different time zones. Using a graphical interface, it displays real-time, auto-refreshing clocks for various global locations with accurate time zone handling. This project offers practical experience in developing real-time applications with Python GUI components.

<u>TABLE OF CONTENTS</u> :-

INTRODUCTION :-

In today's connected world, tracking time across various regions is crucial for business, communication, and travel. This project creates a simplified world clock using Python. It leverages time zone libraries and a graphical user

interface to display live, up-to-date times in multiple cities. The solution enhances understanding of GUI programming, time zone management, and real-time data updates.

OBJECTIVES :-

- Build an easy-to-use world clock application in Python.
- Correctly handle and display times for different time zones with the `pytz` library.
- Design a visually clear, interactive GUI using `tkinter`.
- Refresh displayed times every second automatically.
- Manage errors gracefully, such as invalid time zones.

SCOPE OF THE PROJECT:-

This project serves as a foundation for more advanced time-related applications. It currently supports basic display of multiple cities' times and can be extended into mobile apps, web-based clocks, or integrated calendar tools.

SOFTWARE & HARDWARE REQUIREMENTS:-

Software:

- Python 3.8 or above
- `pytz` library for time zones
- `tkinter` for GUI
- IDEs like VS Code, PyCharm, or IDLE
- Compatible operating systems (Windows, macOS, Linux)

TESTING :-

Test Case	Expected Result	Status
Valid time zones	Correct times shown	Passed
Invalid time zone entry	Shows "Invalid Zone"	Passed
Time refresh every 1 sec	Time updates smoothly	Passed
Display multiple cities	All cities visible	Passed
Window resize handling	Layout remains intact	Passed
Application closes properly	Exits without error	Passed

LIMITATIONS :-

- Fixed list of cities, no user addition feature yet.
- No analog clock display option.
- No support for theme switching (e.g., dark/light modes).
- Not designed for mobile platforms.

FUTURE ENHANCEMENTS :-

- Add city search and dynamic city adding/removing.

- Include option to toggle between analog and digital clock styles.

- Implement dark/light mode themes.

- Allow users to save favorite cities for quick access.

- Integrate real-time APIs for offsets and daylight saving handling.

- Develop mobile and web app versions.

- Incorporate alarms or reminders linked to world times.

CONCLUSION :-

This project successfully developed a functional, user-friendly world clock application in Python, highlighting GUI design, time zone handling, and real-time updating techniques. The foundation laid here supports easy expansion for more sophisticated time management tools.

REFERENCES:-

- Python official documentation

- Pytz library documentation

- Tkinter tutorials and references

<u>APPENDIX (Time Display Sample</u>):-

```
=======================================================
        WORLD CLOCK DISPLAY
=======================================================
Date: 24-11-2025

New York: 05:00:00 AM
        24 November 2025

London:   10:00:00 AM
        24 November 2025

Tokyo:    07:00:00 PM
        24 November 2025

...
=======================================================
```

This version reads naturally, clearly expresses technical details, and is structured for easy presentation in a report format.