
System Requirements Specification Index

For

Current Timestamp and Time Zone Conversions

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1	Project Abstract	
2	Business Requirements	
3	Error! Bookmark not defined.	
4	Template Code Structure	
5	Execution Steps to Follow	Error! Bookmark not defined.

Current Timestamp and Time Zone Conversions

System Requirements Specification

1 PROJECT ABSTRACT

A global travel booking system needs accurate timestamp handling to manage flight bookings across multiple time zones. This assignment focuses on implementing current time retrieval and conversion operations using Python's datetime module. Students will build an interactive system for displaying local and destination times, calculating flight durations, and formatting timestamps for user interfaces.

2 BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	<ol style="list-style-type: none">1. System must retrieve and manipulate current timestamps2. Functions must handle time zone conversions for global locations3. All functions require proper documentation (docstrings)4. System must format timestamps appropriately for different regions5. System must provide an interactive interface for users to access functions6. Error handling must validate inputs and handle edge cases

3 CONSTRAINTS

3.1 INPUT REQUIREMENTS

1. Time Zone Formats:
 - Time zone inputs must be valid IANA timezone identifiers
 - Example: ``"America/New_York"`, `"Europe/London"`, `"Asia/Tokyo"```

2. Function Parameters:

- Current time should use system time when not specified
- Time zone strings must be validated before processing
- Format patterns must follow Python's `strftime()` conventions
- Date inputs must follow YYYY-MM-DD format
- Time inputs must follow HH:MM format in 24-hour time

3.2 FUNCTION CONSTRAINTS

1. Function Definition:

- Each function must perform a specific timestamp operation
- Must include docstrings with examples
- Example: ``def get_current_utc_time():``

2. Timestamp Operations:

- Must use Python's ``datetime`` module
- ``get_current_utc_time()``: Get current UTC timestamp
- ``get_current_time_in_timezone(timezone)``: Get current time in specified timezone
- ``convert_timezone(dt, target_tz)``: Convert timestamp to target timezone
- ``format_timestamp(dt, format_string)``: Format timestamp to regional format
- ``calculate_time_difference(timezone1, timezone2)``: Calculate hours between timezones

3. Return Values:

- Functions must return appropriate types (datetime objects, strings, integers)
- Functions should handle invalid inputs gracefully
- Time differences should be returned as dictionaries with hours and minutes

4. Error Handling:

- Functions should validate timezone identifiers
- Handle invalid format strings
- Validate date and time input formats
- Provide clear error messages to users

3.3 OUTPUT CONSTRAINTS

1. Display Format:

- Timestamps should be formatted according to regional conventions
- Example: ``"March 19, 2025 14:30 EDT"`` (US format)
- Example: ``"19 March 2025 14:30 GMT"`` (UK format)
- Time differences should include hours and minutes
- All outputs should be clearly labeled

2. User Interface:

- Menu-driven interface with numeric options
- Clear input prompts with format examples
- Appropriate error messages for invalid inputs
- Organized display of results

4. TEMPLATE CODE STRUCTURE:

1. Current Time Functions:

- ``get_current_utc_time()`` - returns current UTC time
- ``get_current_time_in_timezone(timezone)`` - returns current time in specified timezone

2. Conversion Functions:

- ``convert_timezone(dt, target_tz)`` - converts datetime to target timezone
- ``format_timestamp(dt, format_string)`` - formats datetime with specified pattern

3. Calculation Functions:

- ``calculate_time_difference(timezone1, timezone2)`` - calculates hours between timezones

4. Main Program Function:

- ``main()`` - provides interactive menu for accessing timestamp functions
- Options for viewing current times, planning flights, and comparing time zones

5. EXECUTION STEPS TO FOLLOW:

1. Run the program
2. Select an option from the menu
3. Follow the input prompts to interact with specific functions
4. Observe results and handle any error messages
5. Continue exploring different options or exit the program