COUNTDOWN TIMER DOCUMENTATION

1.PROJECT OVERVIEW :

The Countdown Timer Project is a simple application that allows users to set a time duration and track the countdown until it reaches zero. The project involves UI/UX design (Figma), System Flow & Architecture (Draw.io), and Version Control (GitHub/Git).

2.TOOL USED :

Figma → UI/UX design for the countdown timer interface.

Draw.io → Flowcharts, system design, and logic representation.

Git/GitHub → Version control and project collaboration.

3.UI/UX DESIGN (Figma tool):

1.KEY FEATURES:

Screen 1: Timer Setup

- **Time Picker**: Users can select hours, minutes, and seconds using scrolling pickers.
- **Quick Presets**: Three circular buttons allow setting predefined times (e.g., 10 minutes).
- **Start Button**: A large, prominent button to initiate the countdown.

Screen 2: Active Timer

- **Circular Progress Display**: Shows remaining time in a bold, central position.
- **Alarm Indicator**: Displays the expected completion time (e.g., 2:35 pm).
- **Control Buttons**:

    - **Pause**: Temporarily stop the timer.
    - **Delete**: Cancel the countdown.
- **Minimal Navigation Bar**: Small icons for menu and settings.

2. UI/UX Design Considerations:

- **Color Scheme**: Soft pink/red palette for a calm and modern look.
- **Typography**: Bold numbers for visibility, smaller labels for clarity.
- **User Experience**:
    - Large buttons for easy touch interaction.
    - Minimal distractions with a clean background.
    - Logical navigation with swipe/toolbar options.

4. Deliverables from Figma:

- **Figma File** containing:
    - Timer setup screen.
    - Active countdown screen.
- **Exported Assets**: Buttons, icons, and timer graphics for development.

4.FLOW & LOGIC DESIGN (draw.io Tool) :

<u>Step 1</u> : Create a Flowchart for the countdown timer logic.

Start →Gather requirements → Desgin the layout (Countdown Timer) → webpage layout → Timer display → color, theme and icons of a design → Input Time → Implementation → Develop UI/UX design → Clickable prototype → Execution → Timer → To check whether it is working or not? → Testing → Check (Time > 0?) → Continue → Else Show "Time's Up" → End.

<u>Step 2</u> : Save diagrams as .drawio and export to .png/.pdf.

5.VERSION CONTROL (Git & GitHub):

1. git config –global user.name “<GitHub user name>”
2. git config –global user.email “<GitHub email>”
3. Create a new repository on GitHub
4. Connect local project to GitHub

   ```
   git remote add origin <Repository URL>
   ```

5. Add files:

   ```
   git add README.md
   ```

6. Initialize Git in your project folder:

   ```
   git init
   ```

7. Commit files:

   ```
   git commit -m “First commit”
   ```

8. Adding files:

   ```
   git add .
   ```

9. Commit file again:

   ```
   git commit -m “First file is added”
   ```

10. Main Branch:

    ```
    git branch -M main
    ```

11. Push files to GitHub:

    ```
    git push -u origin main
    ```

12. Creating branches:

    i. Checkout for old branch and move to new branch :

    ```
    git checkout -b <new branch-name>
    ```

    ii. Pushing files to new branch:

    ```
    git push -u origin <branch-name>
    ```

6. Conclusion:

This project demonstrates the **complete workflow** of software development:

- **Design (Figma)** ensures a user-friendly interface.
- **Flow & Logic (Draw.io)** ensures proper planning and understanding of execution.
- **Version Control (GitHub)** ensures project tracking, backup, and collaboration.
