Speechtimer / Speech Timer

1. Purpose of the Application

This system is designed to display the suggested duration of speeches during events or debates on a monitor, iPad, iPhone, or similar device.

A central admin panel controls start, stop, reset, and messages.

The "Speechtimer" screen displays information on an external monitor for speakers or the audience.

The system runs on a local network (no internet required).

Control and display are provided as an HTML application on any device with a modern browser.

2. The Two Applications

speechtimer.html (Display)

- This is the output for the speaker or audience screen (e.g., projector, external display).
- Automatically updates when controlled from the admin panel.
- Multiple instances on different displays show the same time.
- Open in browser: `http://<server-ip>:<port>/`

admin.html (Control Panel)

- This is the admin panel for controlling the timer. It allows:
  - Selection of speech duration (e.g., 2, 5, 10 minutes)
  - Manual input in minutes or seconds
  - Start, stop, reset
  - Message/fullscreen display
  - Time correction (+/-)
  - Open in browser: `http://<server-ip>:<port>/admin`

Control and Display

- Active timers are automatically synchronized across all screens.
- New screens adopt the current status automatically.
- Active buttons (e.g., last time selection) are visually highlighted.

4. Extensions and Customizations

- Times: Additional time buttons can be added or removed in `admin.html`.

- Layout: Colors and appearance customizable via HTML/CSS.
- Messages: Text entries appear immediately on all connected displays.

5. Technical Basis

- Python 3 + Flask + Flask-SocketIO
- HTML, CSS, JavaScript
- Real-time communication via WebSockets
- Fully offline operation in a local network (LAN/WLAN)

6. Network Requirements

- All devices must be on the same LAN/WLAN.
- Firewall must not block the port (e.g., 55055).
- No internet connection required.

7. Installation & Startup (Windows)

- Install Python: https://www.python.org/downloads/
  - During setup: enable "Add Python to PATH" (both checkboxes).
- Install Python packages (in Windows CMD window):
  - pip install flask flask-socketio flask_httpauth eventlet

8. Project Structure

Place all files in one folder (e.g., c:\speechtimer).

It can be any folder; what's important is that app.py is run from this folder.

Multiple instances can be started, but each must use a different port in app.py.

9. Start

```
cd C:\speechtimer
python app.py
```

10. Access

- Display: http://localhost:55055/
- Admin: http://localhost:55055/admin
- LAN: http://<pc-ip>:55055/

11. Username & Password

Admin area access is password protected.

Defined in `app.py` as:

```
users = {
    "admin": "password123"
}
```

12. Change Port (optional)

Default port is 55055.

Change in `app.py`:

```
if __name__ == "__main__":
    socketio.run(app, host="0.0.0.0", port=55055, debug=True)
```

13. Summary

- Web server with Python + Flask
- Admin controls timer, messages & display
- Everything runs synchronously and offline in the local network