Ticketing System

This project generates, prints, and scans tickets with QR codes. It uses Python, Flask, ReportLab, and PyPDF2.

1. Prerequisites

- Python 3.8+ installed on Windows (64-bit).
- Git Bash or PowerShell for OpenSSL (if using HTTPS).
- Wi-Fi network set to **Private** profile on Windows.
- Windows Firewall: allow inbound TCP port 8000 (create an inbound rule for TCP port 8000).
- Important: Ensure your laptop (for QR generation and server) and all scanning devices are connected to the same network during both ticket creation and validation.

2. Clone and Install

1. Clone the repo into your folder:

   ```
   git clone <repo-url> ticketing_system
   cd ticketing_system
   ```

2. Create and activate a virtual env:

   ```
   python -m venv .venv
   .venv\Scripts\activate
   ```

3. Install dependencies:

   ```
   pip install -r requirements.txt
   ```

3. Generate Tickets

1. Edit `config.py` and adjust the number of tickets `N_TICKET`, the name of the event `EVENT` and the paths and ticket dimensions according to your needs.

2. Place your ticket template PDF into the `ticket_templates` folder. Make sure that the name of the template is simply the name of the event, e.g. `oktoberfest25.pdf`

3. Run script:

   ```
   python generate_tickets.py
   ```

4. Check outputs in `events/{EVENT}`:
    - `tickets.db` holds the predefined number of IDs.
    - `qr_codes/` contains PNGs plus `scan_page.png`.
    - `ticket_urls.csv` maps IDs to URLs.

A backup of each event is being created in `bkp/{EVENT}`.

4. Run the Flask Server

1. Create SSL cert and key (ensure `"/CN=192.168.1.18"` matches your `HOST` in config.py):

```
export MSYS_NO_PATHCONV=1
openssl req -x509 -newkey rsa:2048 -nodes \
  -keyout key.pem -out cert.pem -days 365 \
  -subj "/CN=192.168.1.18"
```

2. Start server with HTTPS:

```
python app.py
```

The console should show:

```
* Running on https://0.0.0.0:8000/
```

3. If using `flask run`, add flags:

```
flask run --host=0.0.0.0 --port=8000 --cert=cert.pem --key=key.pem
```

5. Scanner Webpage

- URL: `https://192.168.1.18:8000/scan`. A separate QR code for accessing this web page is being created in `events/{EVENT}/qr_codes`.
- Page uses live camera and ZXing JS to decode QR.
- Message appears large at center.
- 3 seconds delay between scans.

6. User Instructions

1. Connect to the **Private** Wi-Fi (not a guest network).

2. Open your browser (preferably Safari) and go to the scan URL.

3. Accept the self-signed certificate (click **Erweitert** → **Trotzdem fortfahren**).

4. Allow all required permissions, including the page to use the camera.

5. Do not close the browser tab while scanning.

6. If you accidentally close it, clear site data and cache:

   - In browser settings: **Cookies & Websitedaten**, **Cache** → **Löschen**.

7. Re-open the scan URL and allow camera.

7. Troubleshooting

- **Server unreachable**: check LAN IP, firewall, port-forwarding, AP isolation. Try clear the site and data cache of your browser as described in 7.6.
- **Camera not available**: ensure HTTPS, clear old cache, disable ad-blocker.
- **First scan works, then fails**: use the continuous-scan page; do not rely on built-in camera app handoff.
- **Internal Server Error**: Ask Daniel for help. Something might be wrong with the code.

8. Entrance Staff User Manual

This guide is for staff at the event entrance. Follow these steps to scan and validate tickets smoothly.

1. **Prepare your device**:
   - Make sure your phone or tablet is connected to the event Wi-Fi.
   - Scan the `scan_page.png` QR Code and open the web page in your browser (preferable Safari). You can find this QR code in `events/{EVENT_NAME}/qr_codes`.
   - Keep this tab open; do not close or reload it during the event. In case you close and it does not re-open, clear your browser's data (history and cache).
2. **Grant permissions**:
   - When prompted, accept the security warning for the certificate.
   - Allow the page to access your device camera.
3. **Start scanning**:
   - Point the live camera view at the guest’s QR code.
   - The page will automatically detect the code and display a large message:
   - Green text: “Ticket gültig” → Allow entry.
   - Red text: “Ticket bereits verwendet” or “Ungültiges Ticket” → Deny entry.
4. **Observe buffer time**:
   - After a scan, wait for the message to revert (“Scannen Sie Ihr Ticket”) before scanning the next ticket—about 3 seconds.
5. **Troubleshoot on the spot**:
   - If the page freezes or camera stops: clear this tab’s cache only (in browser settings), then re-open the scan URL.
   - If a QR won’t scan: ensure the code is fully visible and well-lit; tilt device slightly to reduce glare.
   - For any other issue, contact the technical supervisor immediately.
6. **End of shift**:
   - At the end of the event, leave the scan page open until all guests have entered.
   - Do not close the tab until scanning is complete.