RWTH HPC: Run "Speemo" UI and Submit Slurm Training Jobs

What runs where?

- **Terminal 1**: small *interactive* job that only hosts the Flask UI (CPU partition `c23ms` recommended).
- **Training jobs**: submitted by the UI with `sbatch` to GPU partition `c23g` (run independently on nodes like `n23gXXXX`/`w23gXXXX`).

1) Terminal 1 — start the UI on a compute node

```
# Start a small CPU allocation (saves GPUs for training)
salloc -p c23ms --cpus-per-task=2 --mem=4G --time=04:00:00 --constraint=Rocky9

# Once connected to the allocated node:
cd /hpcwork/$USER/speemo_md_0.005
pkill -f "python3 /workspace/run.py" 2>/dev/null || true
./runserve.sh
```

Leave Terminal 1 open; this is your running UI server. To see which node hosts it:

```
squeue -u $USER    # look for the "interact" job; note NODELIST (e.g., w23g0007)
```

2) Terminal 2 — open a browser tunnel to the UI

From your local machine (replace `NODE`):

```
ssh -N -4 \
  -J mk696056@login23-g-1.hpc.itc.rwth-aachen.de \
  -L 127.0.0.1:5000:127.0.0.1:5000 \
  mk696056@NODE
```

Then browse `http://127.0.0.1:5000`.

If you see "Access denied by pam_slurm_adopt" you tunneled to the wrong node. Ensure `NODE` *matches the node of your* interactive *job from* `squeue -u $USER`.

3) Submit training from the UI

Open the UI, set hyperparameters and checkpoint, click **Train**. The server creates an `sbatch` script from your template and submits a GPU job to `c23g`.

4) Monitor jobs and logs

```
# All jobs (interactive + training)
squeue -u $USER

# Live training log (replace JOBID)
tail -f /hpcwork/$USER/logs/train.JOBID.out

# Accounting summary after completion
sacct -j JOBID -X -o JobID,State,Elapsed,NodeList
```

5) Cancel jobs

```
scancel JOBID
```

Notes

- Seeing "two jobs" is normal: one for the UI (interactive) and one per training run (GPU).
- Log files are written by Slurm as `/hpcwork/$USER/logs/train.%J.out` and `.err` (%J = Slurm JobID).