

Event staff manager

- Sam Depauw
- Viktor Walravens
- Xander Vanderhaeghe
- Louis De Veirman
- Casper De Meester



Groep 30

Agenda

Situation & problem

User roles

Our features

Demonstration

Questions



Situation & Problem



User Journey - Manager



LOG IN



NEW EVENT



ACCEPT/REFUS
E



CALENDAR
OVERVIEW



STATISTICS



User Journey – Employee



LOG IN



AVAILABILI
TY



APPLICATION



CALENDAR
OVERVIEW



Our Features

- Statistics
- Ical
- ...



Our Algorithm

Hard Constraints:

1. No overlapping shifts per employee
2. Minimum 1-hour break between shifts
3. Maximum 12 hours per employee per day
4. Availability window constraints
5. Shift capacity must be met

Soft Constraints:

1. Fairness: even hour distribution (60% weight)
2. Availability match: shifts in preferred windows (40% weight)

= HARD CONSTRAINTS =====

```
overlap_constraint(employee_id: str,
                   shift: Dict[str, List[str]],
                   current_assignments: Dict[str, List[str]],
                   all_events: List[Dict]) -> Tuple[bool, Optional[str]]
```

Constraint 1: Employee cannot work overlapping shifts

```
(is_valid, error_message)
```

Employee cannot work overlapping shifts

```
ids = current_assignments.get(employee_id, [])
```

Shift objects

```
= [e for e in all_events if e.get('id') in assigned_shift_ids]
```

Overlaps with new shift

```
if shift in assigned_shifts:
    overlap(shift, assigned_shift):
        assigned_shift_title = assigned_shift.get('title', 'Unknown')
        return False, f"Overlaps with {assigned_title}"
```

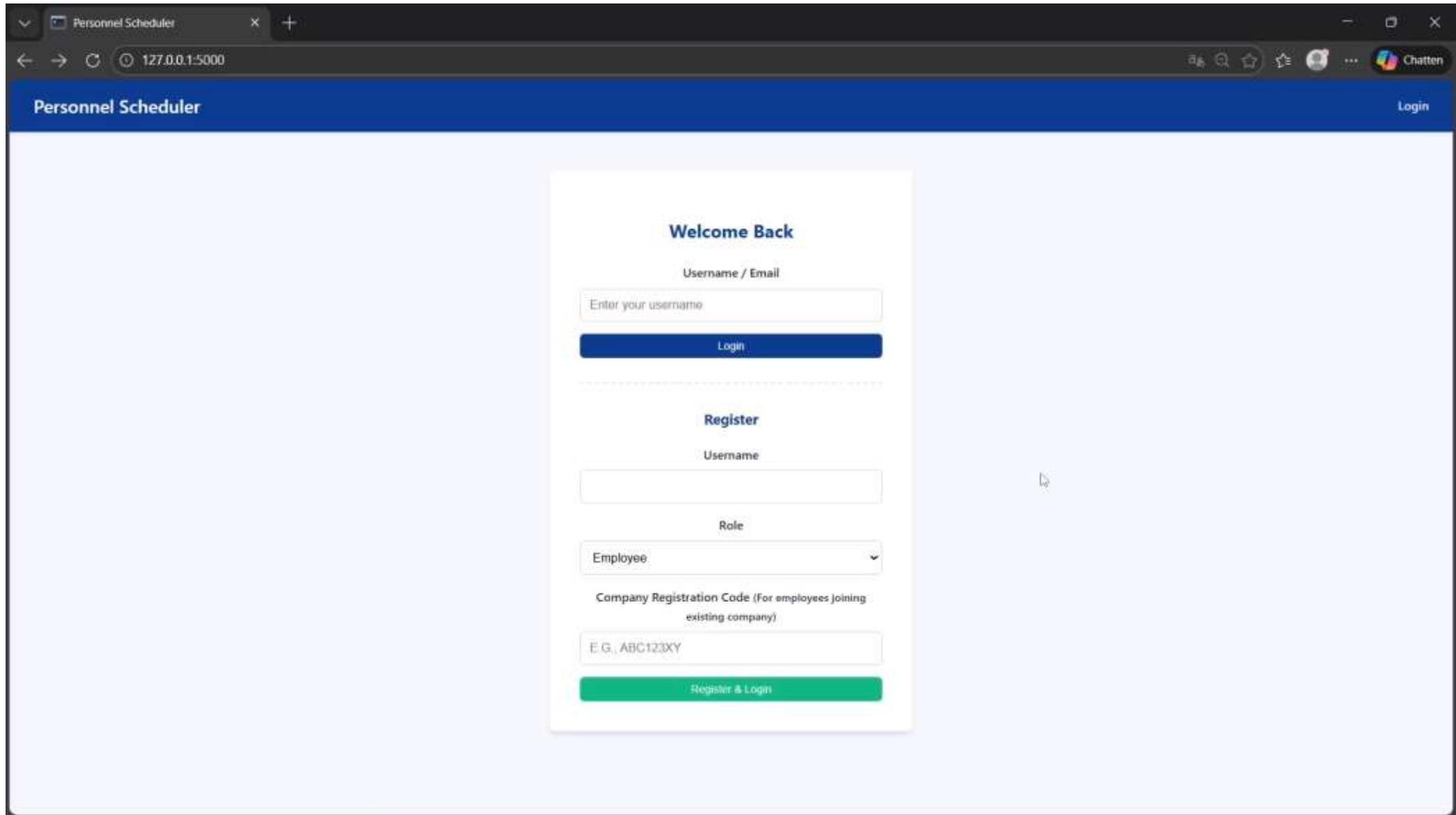
One

```
time_constraint(employee_id: str,
                 shift: Dict[str, List[str]],
                 current_assignments: Dict[str, List[str]],
                 all_events: List[Dict],
                 min_break_hours: float = 1.0) -> Tuple[bool, Optional[str]]
```

Constraint 2: Minimum break time between shifts (default 1 hour)

```
(is_valid, error_message)
```

Demonstration





Questions?

