

Änderungsbericht: Ämtchen Slot Machine

Datum: 4. Dezember 2025

Bearbeitet von: DeepAgent



Zusammenfassung der Änderungen

Alle angeforderten Änderungen wurden erfolgreich implementiert:

1. **Michelle als neue Mitarbeiterin hinzugefügt**
2. **Neue Aufgabe "KaffEEKapseln entsorgen" hinzugefügt**
3. **E-Mail-Adresse auf daria@schnyder-werbung.ch geändert**
4. **E-Mail-Versand-Konfiguration analysiert und dokumentiert**



Detaillierte Änderungen

1. Michelle als neue Mitarbeiterin hinzugefügt

Geänderte Datei: `index.html` (Zeile 169)

Änderung:

```
{ name: "Michelle", schedule: { montag:true, dienstag:true, mittwoch:true,
donnerstag:true, freitag:true }, ferien:false, previous:null }
```

Beschreibung:

- Michelle wurde zur Mitarbeiterliste hinzugefügt
- Standard-Arbeitszeiten: Montag bis Freitag (vollständige Woche)
- Sie wird jetzt in die automatische Aufgabenrotation einbezogen
- Erscheint im UI unter "Mitarbeitende & Anwesenheit"

2. Neue Aufgabe "KaffEEKapseln entsorgen" hinzugefügt

Geänderte Dateien:

- `index.html` (Zeile 154)
- `board.html` (Zeile 87)
- `README.md` (Zeile 14)

Änderungen:

a) index.html - Aufgabenliste:

```
const tasks = [
  "Küche Ordnung",
  "Recycling Dosen",
  "Recycling PET",
  "Karton mit Wendy absprechen",
  "Chef de Frigor",
  "Pausenraum putzen",
  "Getränke auffüllen",
  "Kaffeekapseln entsorgen" // NEU
];
```

b) board.html - Icon-Mapping:

```
const iconMap = {
  // ... andere Icons
  "Kaffeekapseln entsorgen": "delete", // NEU - Material Icon
  "FREI": "sentiment_satisfied"
};
```

c) README.md - Dokumentation aktualisiert:

Die neue Aufgabe wurde zur Liste der verfügbaren Ämtchen hinzugefügt.

Beschreibung:

- Die Aufgabe "Kaffeekapseln entsorgen" ist nun Teil der Rotation
- Verwendet das Material Icon "delete" für die visuelle Darstellung auf dem Board
- Wird wie alle anderen Aufgaben zufällig den Mitarbeitenden zugewiesen

3. E-Mail-Adresse geändert

Geänderte Datei: `send_reminder.py` (Zeile 30)

Änderung:

```
# VORHER:
mail_to = os.environ.get("MAIL_TO", "dario@schnyder-werbung.ch")

# NACHHER:
mail_to = os.environ.get("MAIL_TO", "daria@schnyder-werbung.ch")
```

Beschreibung:

- Die Standard-E-Mail-Adresse für die monatlichen Erinnerungen wurde von "dario" auf "daria" geändert
- Die E-Mail-Adresse kann weiterhin über die Umgebungsvariable `MAIL_TO` überschrieben werden

E-Mail-Versand-Konfiguration

Aktueller Status

Das System verwendet **SMTP** für den E-Mail-Versand über GitHub Actions. Die Konfiguration ist grundsätzlich korrekt implementiert, aber **es fehlen die erforderlichen Secrets in GitHub**.

Erforderliche Konfiguration

1. GitHub Secrets einrichten

Um den E-Mail-Versand zu aktivieren, müssen folgende **Secrets** in den GitHub Repository-Einstellungen hinterlegt werden:

Pfad: Repository → Settings → Secrets and variables → Actions → New repository secret

Secret Name	Beschreibung	Beispielwert
SMTP_HOST	SMTP-Server-Adresse	smtp.gmail.com oder smtp.office365.com
SMTP_PORT	SMTP-Port (optional)	587 (Standard) oder 465
SMTP_USER	SMTP-Benutzername	ki@schnydaer.ch
SMTP_PASS	SMTP-Passwort/App-Passwort	**** (wird verborgen)

Optional (bereits mit Defaults versehen):

```
| Secret Name | Standard | Überschreibbar |
|-----|-----|-----|
| MAIL_FROM | ki@schnydaer.ch | Ja, über Secret |
| MAIL_TO | daria@schnyder-werbung.ch | Ja, über Secret |
```

2. SMTP-Provider-spezifische Anforderungen

Option A: Gmail

- SMTP_HOST: smtp.gmail.com
- SMTP_PORT: 587
- **Wichtig:** Es muss ein **App-Passwort** erstellt werden:
 1. Google-Konto aufrufen
 2. Sicherheit → 2-Faktor-Authentifizierung aktivieren
 3. App-Passwörter → Neues App-Passwort erstellen
 4. Dieses Passwort als SMTP_PASS verwenden

Option B: Microsoft 365/Outlook

- SMTP_HOST: smtp.office365.com
- SMTP_PORT: 587
- SMTP_USER: vollständige E-Mail-Adresse
- SMTP_PASS: reguläres Passwort oder App-Passwort

Option C: Anderer Provider

Kontaktieren Sie Ihren E-Mail-Provider für die spezifischen SMTP-Einstellungen.

3. Workflow-Konfiguration

Die GitHub Actions Workflow-Datei (`.github/workflows/monthly_reminder.yml`) ist bereits korrekt konfiguriert:

```
- name: Send reminder email
env:
  SMTP_HOST: ${ secrets.SMTP_HOST }
  SMTP_PORT: ${ secrets.SMTP_PORT }
  SMTP_USER: ${ secrets.SMTP_USER }
  SMTP_PASS: ${ secrets.SMTP_PASS }
run: python send_reminder.py
```

Zeitplan:

- Automatischer Versand am **1. Tag jeden Monats um 09:00 UTC** (11:00 MEZ / 10:00 CET)
- Manuelle Auslösung über GitHub Actions UI möglich (workflow_dispatch)

Fehlende Konfigurationen und mögliche Probleme

Kritisch - E-Mail-Versand funktioniert nicht ohne diese Schritte:

1. GitHub Secrets fehlen

- **Problem:** Die Umgebungsvariablen `SMTP_HOST`, `SMTP_USER` und `SMTP_PASS` sind nicht gesetzt
- **Auswirkung:** Das Python-Script wirft einen `ValueError` und bricht ab
- **Lösung:** Secrets in GitHub Repository-Einstellungen hinterlegen (siehe oben)

2. SMTP-Authentifizierung

- **Problem:** Reguläre Passwörter werden von modernen E-Mail-Providern oft blockiert
- **Lösung:** App-spezifische Passwörter verwenden (insbesondere bei Gmail)

3. Firewall/Netzwerk

- **Problem:** GitHub Actions Runner könnten von manchen SMTP-Servern blockiert werden
- **Lösung:** Bei Problemen alternativen SMTP-Service testen (z.B. SendGrid, Mailgun)

Empfehlungen zur Verbesserung:

1. E-Mail-Versand testen

```
bash
# Manuell über GitHub Actions UI auslösen:
Actions → Monthly Reminder Email → Run workflow
```

2. Logging verbessern

- Aktuell gibt es keine Bestätigung über erfolgreichen Versand
- Empfehlung: `print()`-Statements im Python-Script hinzufügen

3. Fehlerbehandlung

- Bei SMTP-Fehlern sollte der Workflow einen aussagekräftigen Fehler anzeigen
- Empfehlung: Try-Except-Block um SMTP-Code erweitern

4. Alternative E-Mail-Versand-Methoden

- GitHub Actions bietet keine nativen E-Mail-Funktionen
- Bei anhaltenden Problemen: Externe Services wie SendGrid, Mailjet oder AWS SES verwenden

✓ Checkliste für die Inbetriebnahme

- [] GitHub Secrets für SMTP konfigurieren (SMTP_HOST, SMTP_USER, SMTP_PASS)
 - [] App-Passwort für den E-Mail-Account erstellen (falls Gmail/Outlook)
 - [] Workflow manuell testen über GitHub Actions UI
 - [] Ersten Test-E-Mail-Empfang bestätigen
 - [] Bei Problemen: GitHub Actions Logs überprüfen (Actions → Workflow Run → send_email Job)
-

Zusätzliche Hinweise

Mitarbeiteranzahl vs. Aufgabenanzahl

Aktuelle Situation:

- **12 Mitarbeitende** (inkl. Michelle)
- **8 Aufgaben** (inkl. Kaffeekapseln entsorgen)

Auswirkung:

- Bei voller Besetzung bekommen 4 Mitarbeitende "FREI" zugewiesen
- Die Logik vermeidet, dass jemand zweimal hintereinander dieselbe Aufgabe bekommt
- Mitarbeitende in Ferien bekommen automatisch "FREI"

Besondere Logik

- **"Karton mit Wendy absprechen"**: Wird nur an Mitarbeitende vergeben, die am Mittwoch arbeiten
 - **Anwesenheitstage**: Können pro Mitarbeitender individuell angepasst werden
 - **Persistent Storage**: Zuweisungen werden in localStorage gespeichert für Board-Ansicht
-

Nächste Schritte

1. **Commits pushen** ✓ (wird gleich durchgeführt)
 2. **GitHub Secrets konfigurieren** (manuell im GitHub Repository)
 3. **E-Mail-Versand testen**
 4. **Optional**: Weitere Aufgaben oder Mitarbeitende hinzufügen
-

Ende des Berichts