OutStaff Skills Challenge

Overview

Build new features for the OutStaff workforce management application. You have **2 hours** to complete as many tasks as possible.

Important: You are NOT expected to complete all tasks. This assessment is intentionally designed with more work than can reasonably be finished in 2 hours. We want to see how you approach problems, navigate unfamiliar code, and make progress - not whether you can rush through everything. Do your best work on the tasks you attempt.

Use AI & Internet - We Encourage It!

This is a real-world skills assessment. In real work, you use every tool available to you - so please do the same here:

- **AI Assistants**: ChatGPT, Claude, GitHub Copilot, Cursor, or any other AI tool
- **Search Engines**: Google, Stack Overflow, documentation sites
- **Code References**: Any public repositories, tutorials, or examples

Using these tools effectively IS the skill we're evaluating. Don't hesitate - use whatever helps you work best.

Rules

- **Time Limit**: 2 hours total (hard stop)
- **Tech Stack**: Use the existing stack only (Flask, SQLAlchemy, Jinja2, Tailwind CSS)
- **Database**: Create proper migrations for any model changes
- **Order**: Complete tasks in order (Task 1, then Task 2, etc.)
- **Verification**: Raise your hand when you complete a task. A proctor will verify it works.

Getting Started

The application should already be running. If you need to restart:

```
outstaff restart
```

Before starting tasks, verify you can:
1. Access the app in your browser with the app link
2. create an account
3. Create/access an organization

Tasks

Task 1: Team Notes [Easy]

Create a notes feature for organizations where team members can share information.

Requirements:

- New page accessible from the organization
- Anyone in the organization can add a note
- Notes display with the author's name and timestamp
- Authors can delete their own notes (but not others' notes)
- Notes should persist after page refresh

Task 2: Team Directory [Easy]

Create a directory page showing all members of an organization.

Requirements:

- Display all organization members
- Show each member's name, email, and role
- Admins should be visually distinguishable from regular members
- Include a search/filter to find members by name

Task 3: Expense Tracker [Medium]

Create an expense tracking feature for organizations.

Requirements:

- New Expense model with: amount, category, date, description
- Form to submit expenses with fields for each attribute
- List showing all submitted expenses
- Amounts should display with currency formatting (e.g., "$100.00")
- Display the total amount of all expenses
- Data should persist after page refresh

Task 4: Leave Requests [Medium]

Create a leave request system with an approval workflow.

Requirements:

- Members can submit leave requests (type, start date, end date, reason)
- Leave types: Vacation, Sick, Other
- New requests start with "Pending" status
- Status should be clearly visible (badge or label)
- Admins see approve/reject buttons; members should NOT see these buttons
- Clicking approve changes status to "Approved"
- Clicking reject changes status to "Rejected"
- Data should persist after page refresh

Task 5: Activity Log [Hard]

Create an activity log showing recent actions in the organization.

Requirements:

- Page showing recent activities
- Display what action occurred, who did it, and when
- New actions should appear in the log

- Show activities in reverse chronological order (newest first)

Task 6: Data Export [Medium]

Create a feature to export organization data to CSV.

Requirements:

- User can select what type of data to export
- Clicking export downloads a CSV file
- CSV should have proper headers
- CSV should contain the actual data

Task 7: External Data [Hard]

Fetch and display data from an external API.

Requirements:

- Create a page that fetches data from: `https://jsonplaceholder.typicode.com/users`
- Show a loading indicator while data is being fetched
- Display the external users' names and emails
- Data should be displayed in an organized format (list, cards, or table)
- Page should load smoothly without freezing or crashing
- Page should work correctly on refresh

Task 8: Dashboard Widgets [Hard]

Create a dashboard with multiple information widgets.

Requirements:

- Dashboard page with at least 3 distinct widgets
- Each widget shows different information (e.g., counts, stats, recent items)
- At least one widget should display a number or count
- Widgets should be visually distinct (cards, borders, etc.)
- Layout should be organized (grid or similar)

- Styling should be consistent with the rest of the application

---

Tips

- **Use AI freely** - Ask ChatGPT/Claude to explain code, generate boilerplate, or debug issues
- **Search when stuck** - Google error messages, look up Flask patterns, check documentation
- Read the existing codebase to understand patterns before coding
- Check `app/models.py` for database model examples
- Check `app/templates/` for template patterns
- Use Tailwind CSS classes consistent with existing styles
- Test your features before raising your hand

Good luck!