InstaSched

Wellness Generator

1. Introduction

- **Problem Statement:**
  Many people struggle to carve out short, meaningful wellness breaks in busy days. They know they need to meditate, stretch, or take a walk, but scheduling these into existing commitments can be a hassle.

- **Solution Overview:**
  WellnessGen is a web application that asks users for their daily start/end times, any busy periods, and their lifestyle preferences—and then auto-fills the gaps with a randomized, non-repeating mix of wellness and productive activities. It even lets you export your personalized plan straight into your native calendar!

2. Key Features

1. **Multi-Page Experience:**
    - **Home Page:** Hero banner, elevator pitch, “Get Started” button.
    - **Build Schedule:** Two-column form—time inputs on the left, a two-column checklist of 30+ activity tags on the right.
    - **Activity Library:** Browse every available activity, with cards showing name, duration, and tag badges.
    - **Generated Schedule:** Grid of colorful cards with time blocks and activity names, plus a Chart.js pie-chart summary of time allocation.
2. **Personalization:**
    - **Tag Filtering:** Only schedule activities matching user-selected preferences (e.g. “mindfulness,” “cardio,” “productivity”).
    - **No Repeats:** Ensures every scheduled slot features a unique activity.
    - **Randomized Mix:** Short and long tasks are sprinkled throughout the day for variety.
3. **Calendar Export:**
    - One-click download of a standard `.ics` file containing each time block as a calendar event.
    - Compatible with Google Calendar, Apple Calendar, Outlook, etc.

3. Architecture & Technology Stack

Backend (Flask + Python)

- **Flask** serves four main routes:
  - `/` **(Home)**
  - `/build` **(Build Schedule)** handles GET (show form) and POST (process inputs).
  - `/library` returns the full activity list.
  - `/download.ics` reads the last schedule from `session` and uses the **ics** library to build an iCalendar file.
- **Scheduling Algorithm:**
  - **Parse inputs:** Convert “HH:MM” strings and busy-period ranges into `datetime` objects.
  - **Calculate free periods:** Subtract busy periods from the day window.
  - **Fill slots:** In each free gap, randomly select from activities whose duration fits—excluding any already used—to guarantee uniqueness.
- **Data Model:**
  - `wellness_activities.json` contains 40 items, each with `name`, `duration` (in minutes), and a list of `tags`.
- **Session Storage:**
  - The generated schedule list is saved in Flask’s `session` so that the download route can retrieve it without re-posting form data.

Frontend (HTML/CSS/JS + Bootstrap + Chart.js)

- **Template Inheritance:**
  - `base.html` defines the navbar, footer, and includes Bootstrap CSS/JS, Bootstrap Icons, Chart.js, and our custom `styles.css` and `main.js`.
- **Responsive Layout & Styling:**
  - **Bootstrap 5** provides the grid system and components (cards, buttons, modals, tooltips).
  - Custom CSS adds a light-blue page background, orange accent borders, card shadows, hover transitions, and a two-column preferences grid.
- **Interactive Enhancements:**
  - **Tooltips** on form labels (e.g. “What does this mean?”) initialized in `main.js`.
  - **Pie Chart** showing distribution of minutes per activity, rendered by Chart.js in the schedule page.
  - **Export Button** styled with Bootstrap icon (`bi-calendar2-plus`).

4. Live Demo Walk-Through

1. **Home Page**
    - Show hero image, site branding (“WellnessGen”), and “Get Started” CTA.
2. **Build Schedule**
    - Enter **Start**/**End** times (e.g. 08:00–18:00).
    - Add busy periods (`09:00-10:00, 13:00-14:00`).
    - Select tags (e.g. “mindfulness,” “flexibility,” “productivity”).
    - Click **Generate Schedule**.
3. **Generated Schedule**
    - Observe the **grid of cards**, each colored uniquely and showing its time block and activity.
    - View the **pie chart**: breakdown of total minutes per activity.
    - Click **Export to Calendar** → download **`schedule.ics`** → open in native calendar.
4. **Activity Library**
    - Browse all activities, filter by tag badges, inspect durations.

5. Challenges & Solutions

- **Ensuring No Repeats:** Kept a `used_names` set to filter out already-scheduled activities.
- **Responsive Preferences List:** Used CSS Grid for a fluid two-column checklist that scrolls if it grows too tall.
- **Data Visualization:** Moved Chart.js init to `main.js` and passed data via a `data-chart` attribute to avoid Jinja quoting issues.

7. Conclusion

WellnessGen demonstrates how a simple idea—automating your wellness breaks—can be turned into a polished, user-friendly web app by combining:

- **Clean UI/UX** with Bootstrap and custom styling
- **Robust scheduling logic** in Python/Flask
- **Portable export** via iCalendar
- **Data visualization** for immediate feedback