**Habit Tracker Web Application Documentation**

**Group Members**

1. Felix Mbugua SCT212-0248/2023
2. Cecil Kioko SCT212-0047/2023
3. Shemiramoth Mugo SCT212-0061/2023
4. Lee Kariuki SCT212-0049/2023
5. Samuel Kuria SCT212-0476/2023
6. Dennis Kirehu SCT212-0045/2023

**A Django & JavaScript-Based Productivity Tool**

**Distinctiveness and Complexity**

**Uniqueness of the Project**

This **Habit Tracker** stands out from basic to-do lists or simple task managers because:

* **Visual Habit Tracking**: Uses a calendar-style interface to track daily progress with color-coded completion status.
* **Streak System**: Dynamically calculates and displays streaks with emoji rewards (🔥 for streaks).
* **Interactive UI**: Users can toggle habit completion by clicking calendar days.
* **Mobile-First Design**: Fully responsive, ensuring usability on phones, tablets, and desktops.
* **Django-Powered Backend**: Unlike pure frontend trackers, this integrates a scalable backend for future user authentication and data persistence.

**Complexities Involved**

1. **Dynamic Calendar Generation**: JavaScript generates a calendar based on habit start dates.
2. **Streak Calculation Logic**: Computes the longest streak of completed habits.
3. **Django Integration**: Combines Django templating with JavaScript for a seamless frontend-backend interaction.
4. **LocalStorage Fallback**: Works offline but syncs with Django when available.

**Design Approach**

**Why Django + JavaScript?**

* **Django** provides a structured backend for future expansion.
* **JavaScript** enables a dynamic, interactive frontend without full page reloads.

**UI/UX Decisions**

* **Minimalist Design**: Focuses on usability with clear visual feedback (green = completed, gray = missed).
* **Mobile-First**: Ensures accessibility on smaller screens with touch-friendly buttons.
* **Floating Action Button (FAB)**: A "+" button for quick habit addition, following modern UI trends.

**File Structure & Contents**

**Backend (Django)**

1. models.py
   * Habit: Stores habit name, start date, and user (if authentication is added).
   * HabitCompletion: Tracks daily completions (date + boolean status).
2. views.py
   * Renders the main page (index.html) and handles habit CRUD operations.
3. urls.py
   * Routes URLs to views (e.g., / → habit tracker homepage).
4. settings.py
   * Configures static files, templates, and security settings.

**Frontend**

1. index.html
   * Base template with:
     + Header ("Habit Tracker")
     + Habit cards (dynamically loaded via JavaScript)
     + Modal form for adding habits
     + Mobile-responsive layout
2. styles.css
   * Custom styling for:
     + Habit cards, calendar days, streaks, and buttons
     + Responsive breakpoints (mobile/desktop)
3. script.js
   * Handles:
     + Rendering habits from data
     + Toggling completion status on day clicks
     + Calculating and displaying streaks
     + Saving/loading habits (LocalStorage + Django API)

**How to Run the Application**

**Prerequisites**

* Python 3.8+
* Django 4.0+
* Modern web browser (Chrome, Firefox, Edge)

**Setup Steps – Bash commands**

1. **Clone the repository**:

*git clone [your-repo-url]*

**then**

*cd class\_project*

1. **Set up a virtual environment**:

*python -m venv venv*

**then**

*venv\Scripts\activate*

1. **Install dependencies**:

*pip install django*

1. **Run migrations**:

*Python manage.py makemigrations*

**then**

*python manage.py migrate*

1. **Start the server**:

*python manage.py runserver*

1. **Access the app**:  
   Open <http://127.0.0.1:8000> in your browser.

**Additional Information**

**Key Features**

1. **Add Habits**: Name + start date.
2. **Track Daily Progress**: Click calendar days to toggle completion.
3. **Streak Counter**: Visual feedback with emojis (🔥 = active streak).
4. **Mobile-Friendly**: Works on phones, tablets, and desktops.

**Why This Project?**

This project demonstrates:

* **Full-stack proficiency** (Django + JavaScript)
* **Interactive UI design**
* **Problem-solving** (streak logic, dynamic calendar)