

How to Use S.C.W. E.C.O.

Last Updated: August 2026

[S.C.W. E.C.O.](#)

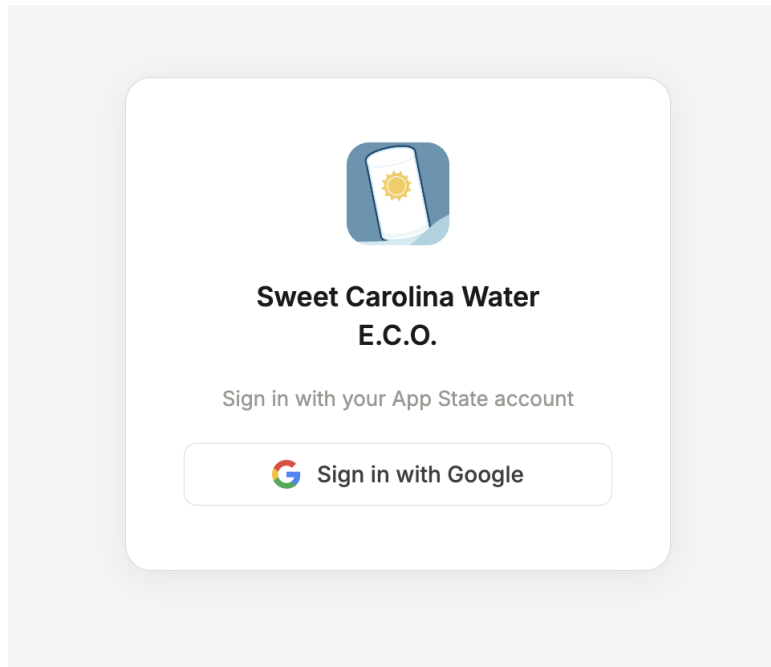
ECO is Sweet Carolina Water's event and content organizer. This application is used to create events, organize sign-ups, and consistently schedule content. ECO helps Sweet Carolina Water and the Association of Student Entrepreneurs by creating efficiency and consistency, which are crucial to successfully promoting.

PASSWORD: SCW2026

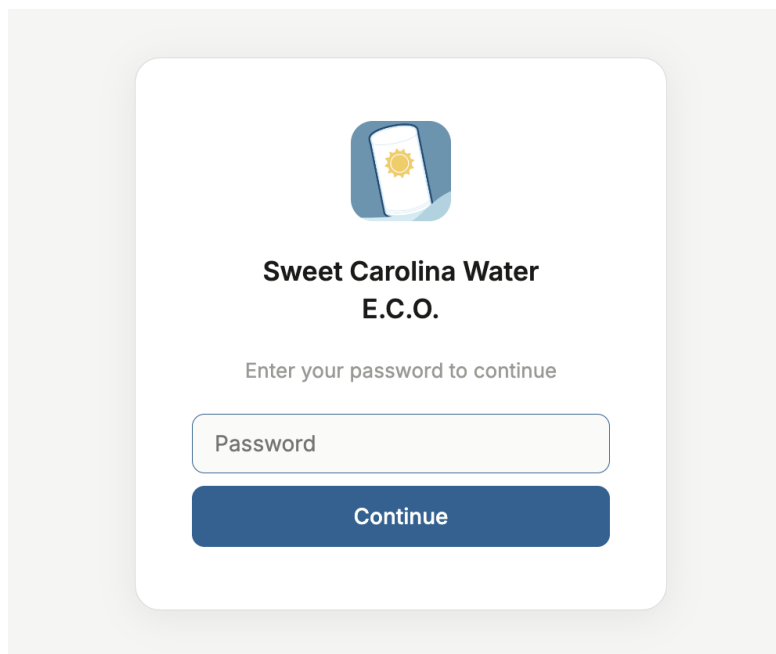


Authentication & Access

- Google OAuth sign-in restricted to [@appstate.edu](#) accounts



- Shared team password as a second layer (24-hour sessions)



- Automatic session persistence — stay logged in across visits

- Automatically logs you out after 24-hours

Event Calendar

- Monthly calendar view with navigation (prev/next month)

August < 2026 >

← Today →

- Add, edit, and delete events with: name, date, type, time, location, and notes

Add event ×

August 10, 2026

Event name

e.g. App State Farmers Market

Added by

Tristan Spinato

Type

Tabling ▼

When

9:00 AM ▼ to 5:00 PM ▼

Where

e.g. Student Union

Notes (optional)

e.g. Extra Details

Save

Cancel

- Volunteer signup tracking per event (add/remove names, see count)

VOLUNTEERS SIGNED UP

Tristan Spinato

4:00 PM – 5:00 PM x

Tristan Spinato

9:00

AM

▼

to

5:00

PM

▼

spinatotg@appstate.edu

Sign up

Edit

Emails

Close

- Event type color-coding with a legend (fully customizable colors)

App State U.S. Holiday Tabling Partnership Volunteer Work Meeting Other

Customize ▼

EVENT TYPES

App State

↑ ↓

U.S. Holiday

↑ ↓

Tabling

↑ ↓ ×

Partnership

↑ ↓ ×

Volunteer Work

↑ ↓ ×

Meeting

↑ ↓ ×

Other

↑ ↓ ×

Type name

+ Add

- U.S. federal holidays displayed automatically
- Day overflow modal for dates with many events
- Reminder email generator — pre-fills a volunteer signup email for any event

Signup Emails

Marketing Meeting

Signed up with emails

Copy Emails

spinatotg@appstate.edu

Email template

Copy Message

Hi Everyone,

This message has been sent out as a reminder that you have signed up for an event!

Name: Marketing Meeting

Date: August 19, 2026

Time: 4:00 PM – 5:00 PM

Location: CFE

Close

- CSV import — bulk import App State calendar events from a spreadsheet

IMPORT APP STATE CALENDAR

Upload a CSV file to bulk-import App State events. Columns: date, name, type, notes . Date format: YYYY-MM-DD . Type must match an existing event type key.

Choose CSV

- Event listing platforms panel (Explore Boone links, etc.) — synced in real time

Event Platforms ▼

Explore Boone

↗ ↑ ↓ ✕

NC Festivals

↗ ↑ ↓ ✕

Name

URL

+ Add

Content Calendar

- Monthly content calendar view with navigation

August < 2026 >

←

Today

→

- Add, edit, and delete social media posts per date

Add post×

August 20, 2026

Name

e.g. Summer tabling recap

Post type

Video

Added by

Tristan Spinato

Caption draft

Write your caption here...

Hashtags

#sweetcarolinawater #studentrun #sustainability #ex

Notes (optional)

e.g. Extra Details

Save

Cancel

- Post status tracking: Pending → Approved → Posted / Scrapped

STATUS

Approved

Posted

Scrapped

- Weekly schedule editor — assign platforms to specific days of the week

PLATFORM SCHEDULE

Sunday		+ Add ▾
Monday	LinkedIn ← → × Instagram ← → × Facebook ← → ×	+ Add ▾
Tuesday	TikTok ← → ×	+ Add ▾
Wednesday	Instagram ← → × Facebook ← → ×	+ Add ▾
Thursday	TikTok ← → ×	+ Add ▾
Friday		+ Add ▾
Saturday	Instagram ← → × Facebook ← → ×	+ Add ▾

Save Schedule

- Default hashtags editor — pre-fill hashtags on every new post

DEFAULT HASHTAGS

#sweetcarolinawater #studentrun #sustainability #exploreboone

Save Hashtags

- Platform type manager — add/recolor/reorder social platforms

Customize ▾

PLATFORM TYPES

LinkedIn	↑ ↓ ×
Instagram	↑ ↓ ×
TikTok	↑ ↓ ×
Facebook	↑ ↓ ×

Platform name + Add

- Day overflow modal for dates with many posts
- Post detail view with full info and status controls

- Customize section with platform legend

Social Platforms ▼

LinkedIn

↗ ↑ ↓ ✕

Instagram

↗ ↑ ↓ ✕

TikTok

↗ ↑ ↓ ✕

Facebook

↗ ↑ ↓ ✕

Name

URL

+ Add

Content Creator Inspiration ▼

Liquid Death

↗ ↑ ↓ ✕

Cleuely

↗ ↑ ↓ ✕

Jun Yuh

↗ ↑ ↓ ✕

Name

URL

+ Add

THE SWEET CAROLINA BRAND

The Sweet Carolina logo is the heart of our brand and should always appear clear, confident, and consistent. Use the primary logo whenever possible, and choose approved color variations only when needed for contrast or readability. Always give the logo breathing room — nothing should crowd, overlap, or compete with it. The logo should never be stretched, squished, recolored, rotated, or decorated with effects. Use official logo files only, selecting SVG or EPS for print and PNG for digital use.

COLORS

Sweet Carolina's colors are bold, refreshing, and inspired by the Carolina mountains and sky. Always use the approved color palette. Avoid unapproved colors or altered shades.



TYPOGRAPHY

Use the approved brand fonts consistently across print and digital. Avoid substituting fonts or altering letterforms.

HEADLINE — TUSKER GROTESK 4600 SEMIBOLD

ABCDEFGHIJKLMNOPQRSTUVWXYZ

SUBHEAD — TUSKER GROTESK 3500 MEDIUM

ABCDEFGHIJKLMNOPQRSTUVWXYZ

BODY — SAHITYA REGULAR / BOLD

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

Schedule Tab

- Social Platforms panel — links to all team social accounts (add/reorder/delete)
- Event Platforms panel — links to event listing sites (Explore Boone, etc.)
- Content Creator Inspiration panel — curated links for content ideas
- Brand Guidelines reference — logo usage, color palette, typography, photography guidelines

Settings

- How to use E.C.O. — link to onboarding guide
- Activity Log — full real-time log of every action taken (who, what, when)
- Team section — team contacts and roles
- Semester Report — date-range analytics: total events, signups, posts by platform/status, most active users
- Personal section — set your display name
- Reminder Email Template editor — customize the volunteer reminder email
- Privacy Policy and Terms of Use modals

Real-Time Sync (Firestore)

- All four data sources sync live across all users simultaneously:
 - Events calendar
 - Content posts

- Settings (event types, platforms, inspiration links)
 - Activity log
-
- No page refresh needed — changes appear instantly for everyone

Cross-Cutting

- Mobile-responsive design with bottom tab bar on small screens
- Dark/light theme personalization

How Netlify Works (Hosting)

Netlify is where the app lives on the internet. When you deploy, it takes your `index.html` file and serves it globally from a CDN (content delivery network). This means the file is cached on servers around the world, so it loads fast from anywhere.

Your setup:

- The app is a single static file (`index.html`) — no server, no backend, no database on Netlify's side
- Netlify just serves that file when someone visits `scweco.netlify.app`
- `netlify.toml` tells Netlify to attach security headers to every response (X-Frame-Options, etc.)
- Deployments are manual — you drag and drop the project folder (or connect a GitHub repo for auto-deploys)
- HTTPS is automatic and free — Netlify provisions and renews the SSL certificate for you
- If the file changes, you redeploy, and the new version is live within seconds

What Netlify does NOT do:

- It doesn't store any of your app's data
- It doesn't run any server-side code
- It doesn't handle authentication — that's Firebase

[Firebase](#) - Database for the platform

How Firebase Works (Data & Auth)

Firebase is Google's backend-as-a-service. E.C.O. app uses two Firebase products:

Firebase Authentication

- Handles the Google OAuth sign-in flow entirely — your app never sees passwords
- When a user clicks "Sign in with Google," Firebase opens the Google consent screen, verifies the account, and returns a signed token
- The app checks that token's email ends in `@appstate.edu` before allowing access
- Firebase keeps the session alive so users don't have to sign in every visit

Firestore (Database)

Firestore is a real-time NoSQL database organized into **collections** → **documents**. Your app uses one collection (`calendar`) with four documents:

How data flows:

1. A user makes a change (adds an event, updates a platform color, etc.)
2. The app writes the new data to Firestore via `set()`
3. Firestore notifies every other connected client via `onSnapshot()` — a persistent listener that fires whenever the document changes
4. Each client receives the new data and re-renders the relevant part of the UI instantly

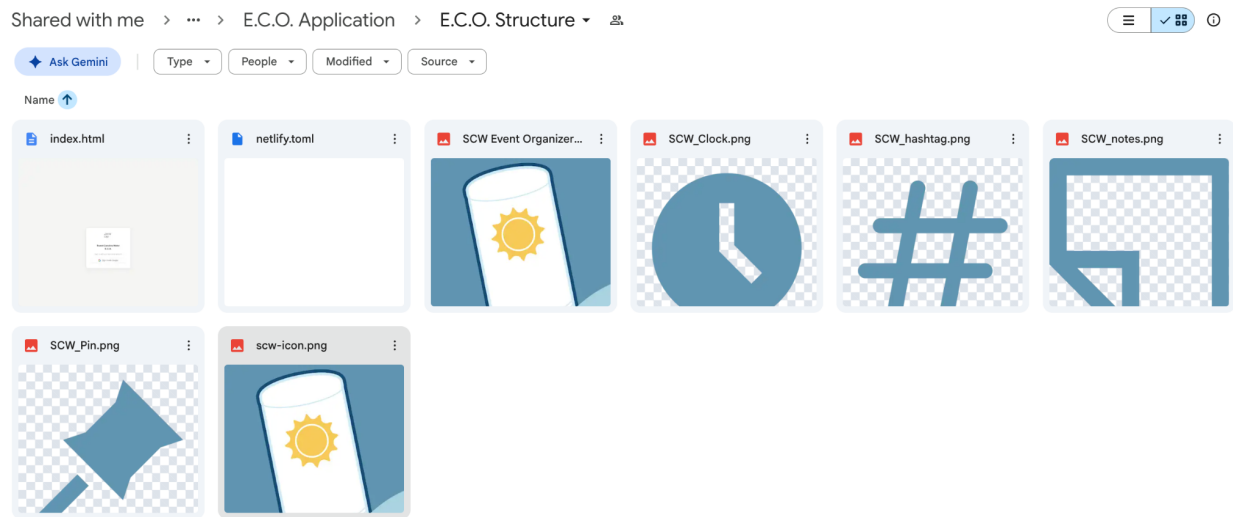
Security rules sit in front of all reads and writes — only signed-in [@appstate.edu](#) accounts are permitted. Anyone without a valid Firebase auth token is blocked at the database level, regardless of whether they somehow access the app file.

What Firebase does NOT do:

- It doesn't host the app (that's Netlify)
- It doesn't send emails
- It doesn't run scheduled tasks or background jobs

Fail-safe

If there is a potential security threat or bug within the code, you will have access not only to Netlify and Firebase, but also to the HTML as well.



In the Sweet Carolina Water Marketing Drive, you will find all assets to the application, including HTML, Netlify TOML, and all assets used.

Marketing & Branding Folder → Websites → E.C.O. Application → E.C.O. Structure

Suggestions on how to update E.C.O.

Working with CS or CIS majors to fix your issues would be great to ensure the internal tool runs smoothly.

Other tools

[Claude](#)

Good for updating and running cleanups on code

[VS Code](#)

Good for taking a look at the code yourself and manually editing it

How to repupload to Netlify:

Step 1 — Make Your Changes

Open the SCW ECO folder. All app changes happen inside index.html. Replace the old one in the folder with the new one.

Step 2 — Log Into Netlify

Go to netlify.com and sign in. On the dashboard, click the scweco site.

Step 3 — Open the Deploys Tab

Click the Deploys tab at the top of the site page.

Step 4 — Drag and Drop to Deploy

Scroll to the bottom of the Deploys page. You will see a box that says:

"Want to deploy a new version of your site? Drag and drop your site output folder here"

Drag the entire SCW ECO folder (not just index.html — the whole folder) into that box. Netlify will upload everything inside it.

Step 5 — Wait for the Deploy to Finish

A new deploy will appear at the top of the list with a yellow "Publishing" status. Wait for it to turn green and say "Published" — this usually takes 15–30 seconds.

Step 6 — Verify the Update

Open scweco.netlify.app in a browser. Do a hard refresh (Cmd + Shift + R on Mac, Ctrl + Shift + R on Windows) to bypass the browser cache and confirm the new version is live.

If Something Goes Wrong — Roll Back

On the Deploys tab, every past deploy is listed with a timestamp. Click any previous deploy, then click "Publish deploy" to instantly roll back to that version. Nothing is ever permanently lost.

Questions?

Contact Tristan Spinato - previous marketing director and designer of the application
spinatotristan@gmail.com