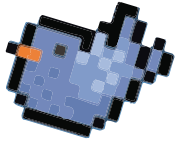


CS4530 Final Project: "User and Town Persistence"

Group 602: Rhyme, Zachary, Iyanna, and Oleksii



Our Feature: User and Town Persistence

In the original version of Covey Town everything is stored in-memory. In our version of Covey Town, we created the concept of a user account and require that all users must create an account. Display name and login information is stored in a database so they are not deleted once the app is shutdown.

We also introduced town persistence. We now persist their ID, name, maximum occupancy, map file, and whether or not they are publicly visible. Additionally we track all towns that a user has previously joined including private. They can easily access their towns from the profile page without having to logout.

Lastly, we created the concept of the "founder" of a town and "residents" of a town. Now, only the founder of a town will see the "Town Settings" menu option and be able to update a town's name and whether or not the town is publicly listed. Residents will not see the "Town Settings" menu option, and the concept of a "town update password" has been removed altogether.

Demo and Source

Our demo site is available at <https://covey-town-webapp-602.onrender.com>, and our code at <https://github.com/neu-cs4530/fall23-team-project-group-602>

Profile Page

Display Name cookie

Email

The Profile Page, users can change display name or join towns they belong to

Edit town cereal (5F26F699)

Friendly Name

Publicly Listed

The Town Settings Page, similar to the original, but we removed town password

Our Tecnology Stack and Design

When users first load the app they are prompted to login or register an account. We used postgres as a database to hold login information. The frontend connects to various backend API's to obtain a userID. We used passport.js to implement authentication and verify login parameters. Users can also login through Google which use react-oauth package to send information to the backend API. Their display name and userID are persisted by various React states. After logging in users can create or join an existing town in the database. Most of the preexisting town code was used for this but towns are now backed by a ModelTown (sequelize model), which is an instance variable in the Town class. When the app is booted up and the TownStore is initialized, all the ModelTowns are loaded from the database, and from them Towns are created and added to the TownsStore. Users can also view their profile once connected to a town. This allows them to change their display name which is persisted to the backend or view and connect to towns they have joined before.

Our Future Work

If we continued to work on persistence integration we have a few frontend and backend changes we would make. In the backend we want to continue persistence by having location persistence in towns so users are spawned at their last logged location. We also want to allow town founders to promote trusted users to town admins who would have more control over the town. Additionally, we want to integrate a forget password screen from the login so users do not have to make a new account if they forgot their password. Lastly, we want to polish up the UI and style the rest of the Covey Town UI in our design system.

Welcome to CoveyTown

Don't have an account yet? [Register Here](#)

Email

Password

Or Login with

The Login Page, already registered users can login with their account info or Google sign-in

Welcome to CoveyTown

Have an account? [Login Here](#)

Display Name *

Email *

Password *

The Registration Page, users are required to set account info and display name, which can later be changed on the profile page