

For the PoC, we added database functionality that supports creating, editing, and deleting templates. Notably, we were able to edit/store/display the blurbs and the template body as HTML content, with a custom rich text editor that Steph built, so the blurbs can include links and the body can include blanks, dropdowns, and optional content (in the form of buttons). We additionally implemented a feature to mangle the email subject and body so that it can find its way around spam filters (dubbed the FilterBuster feature).

We did not implement the concept of recipient bundles. We just specified emails for recipients, but for the MVP and final project we hope to add to these templates by allowing contacts and groups.

We also accomplished simple user logins to store user info, and to log in, sign out, and redirect -- although we don't use cookies so your session ends when you refresh the page. We would eventually like to upgrade the sign up/login to be persistent and more secure, but overall the storage side is implemented.

Similarly, updating user information is not implemented right now. The settings page is available, but users cannot submit the form, nor will it update the backend. However, in terms of the backend this will not be complex as it requires simple api endpoints and update/delete query calls.

We also implemented an initial version of searching and filtering templates by tags, authors, locations, and drafts. Some API routes are non-ideal (calling a POST to get template info, instead of using a GET). These were intentional choices in order to get the application up and running. We will improve these functionalities for later milestones.

It should also be noted that since this is a PoC, we were not as thorough with validation, error prevention, input sanitization, error handling, or polishing the front-end (for example, displaying a spinner icon when templates are loading, or giving appropriate failed request alerts or redirections). These will be added/fine-tuned for the MVP and finished product.

In the upcoming MVP and final deadlines, we will need to have bookmarks, comments, and moderation data stored. We've already started brainstorming how to set up these tables and do not see this as a problem for the future.

Overall, we are happy with what we accomplished as we were able to find a way to have persistent data storage on Heroku (unlike sqlite), and we were able to implement account creation and authentication, as well as template creation, modification, and usage. Thus we feel we made large improvements in the short time to accomplish our PoC.