

Project Features(in priority order):

Profile Page:

- Display user info (name, username, *bio*, profile picture, level, achievements)
- Ability to change account info (name, username, *bio*, *profile picture*)
- (click to view friends list)

Tracking Page:

- Ability to add an item tracked (water bottles, *etc.*)
- When enough items are tracked, the user level updates

Friends Page:

- Ability to view friends' progress (level and number of water bottles tracked - *also name, username and profile picture*)
- Button to search for and add friends (which will then add the friend to the view list)
- List of friends' names and a button to view each individual's profile

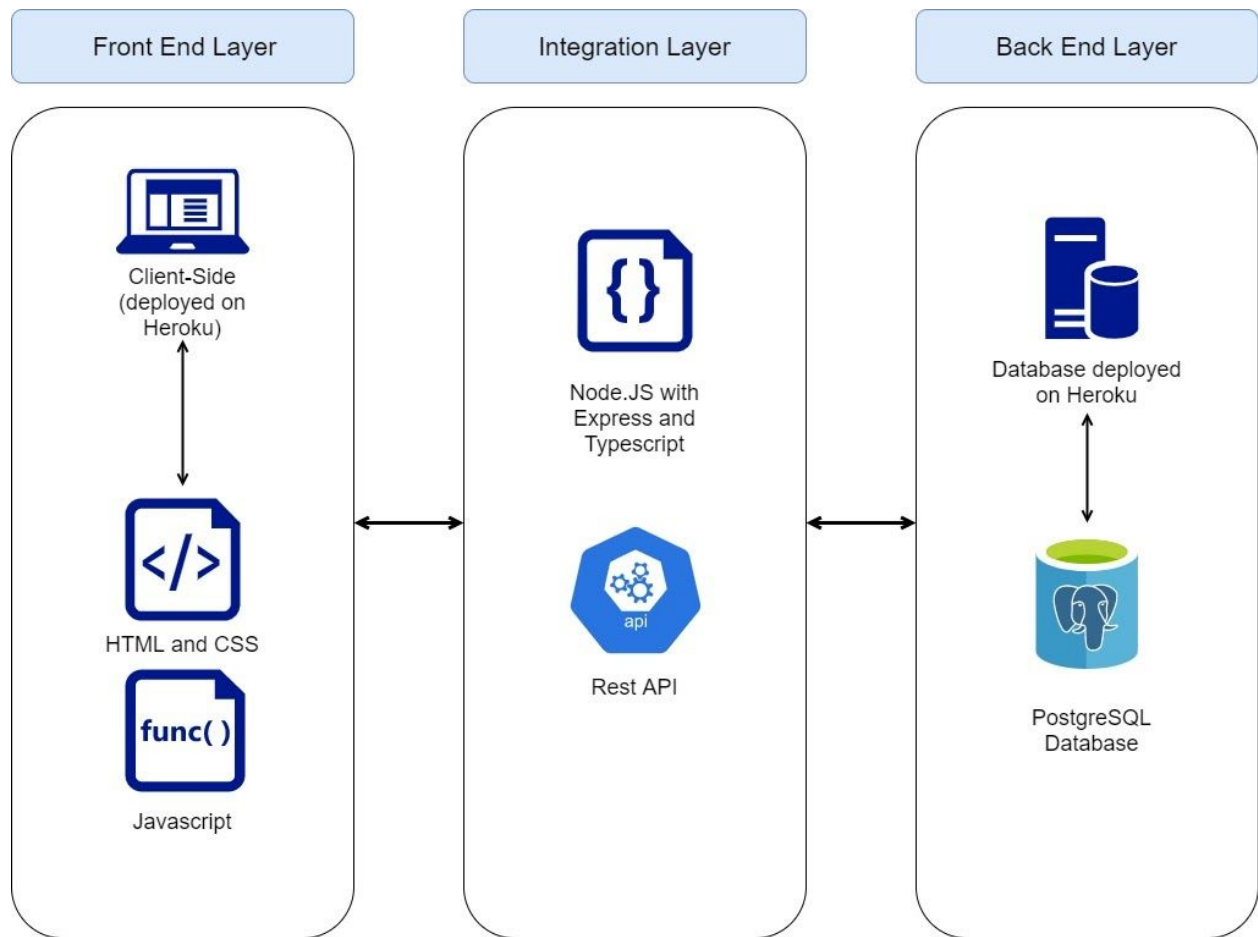
Donate View:

- Button for link to donate to certain websites and resources
- Information about each organization

Home Page:

- Contains links to sign up or login
- Displays information about the web application and what our purpose is

Architecture Diagram:

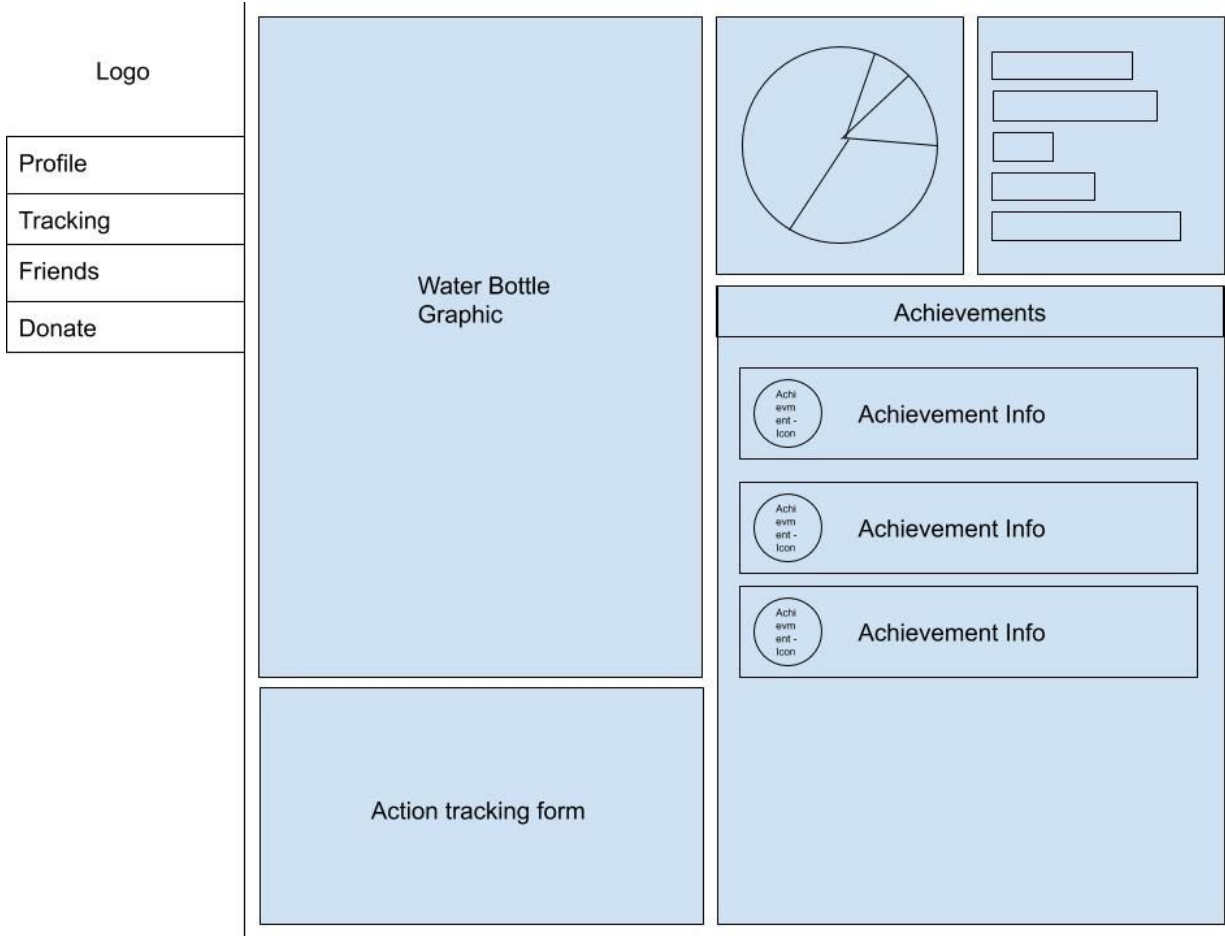


Front-End Design: (wireframes)

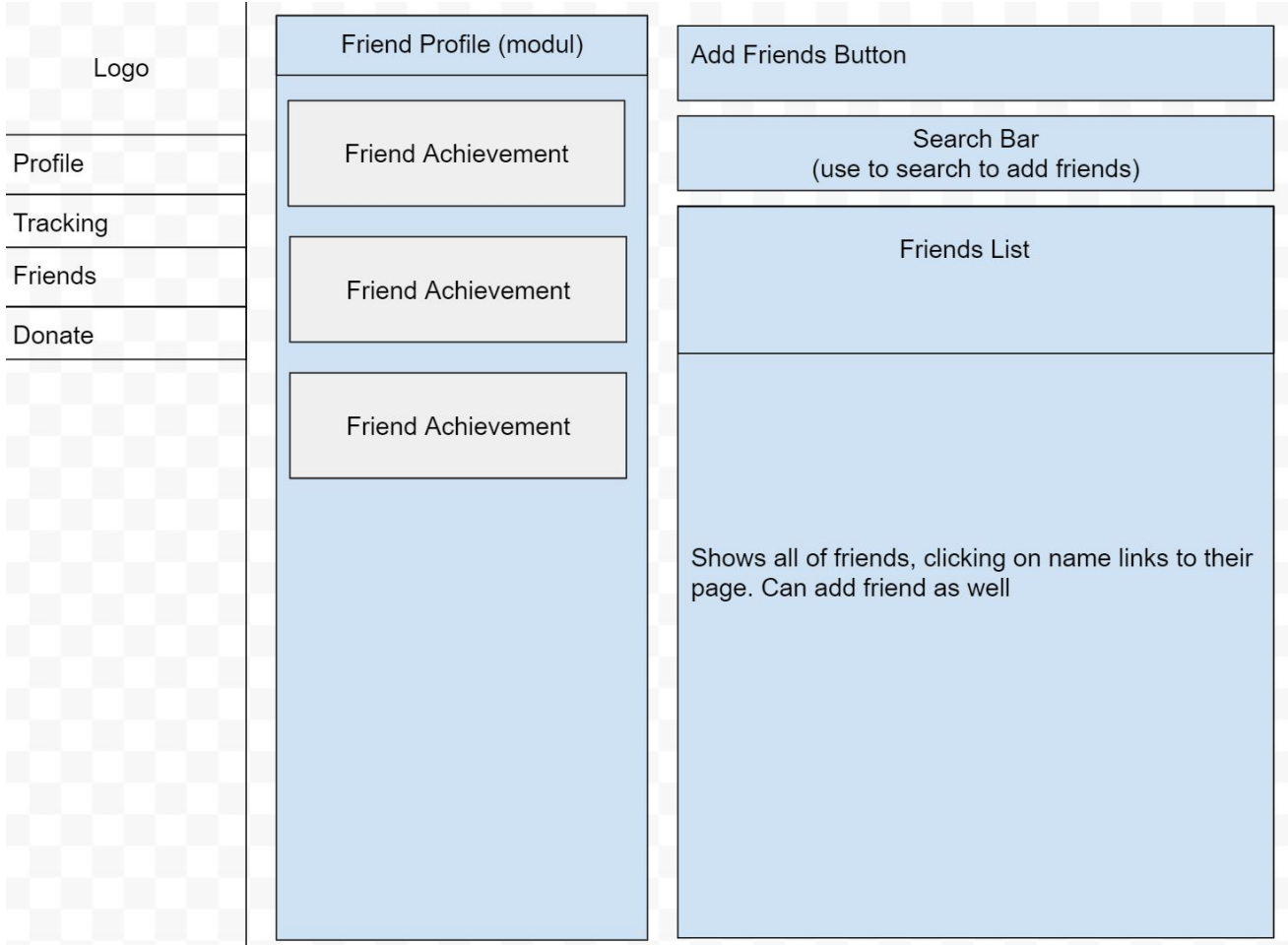
Profile Page

Logo	<div>Profile Pic</div>	Username
Profile		Name
Tracking		Level: sum of achievements
Friends		<div>Update Profile</div>
Donate		
	<div>My Progress</div>	
	<div>Achievements</div>	

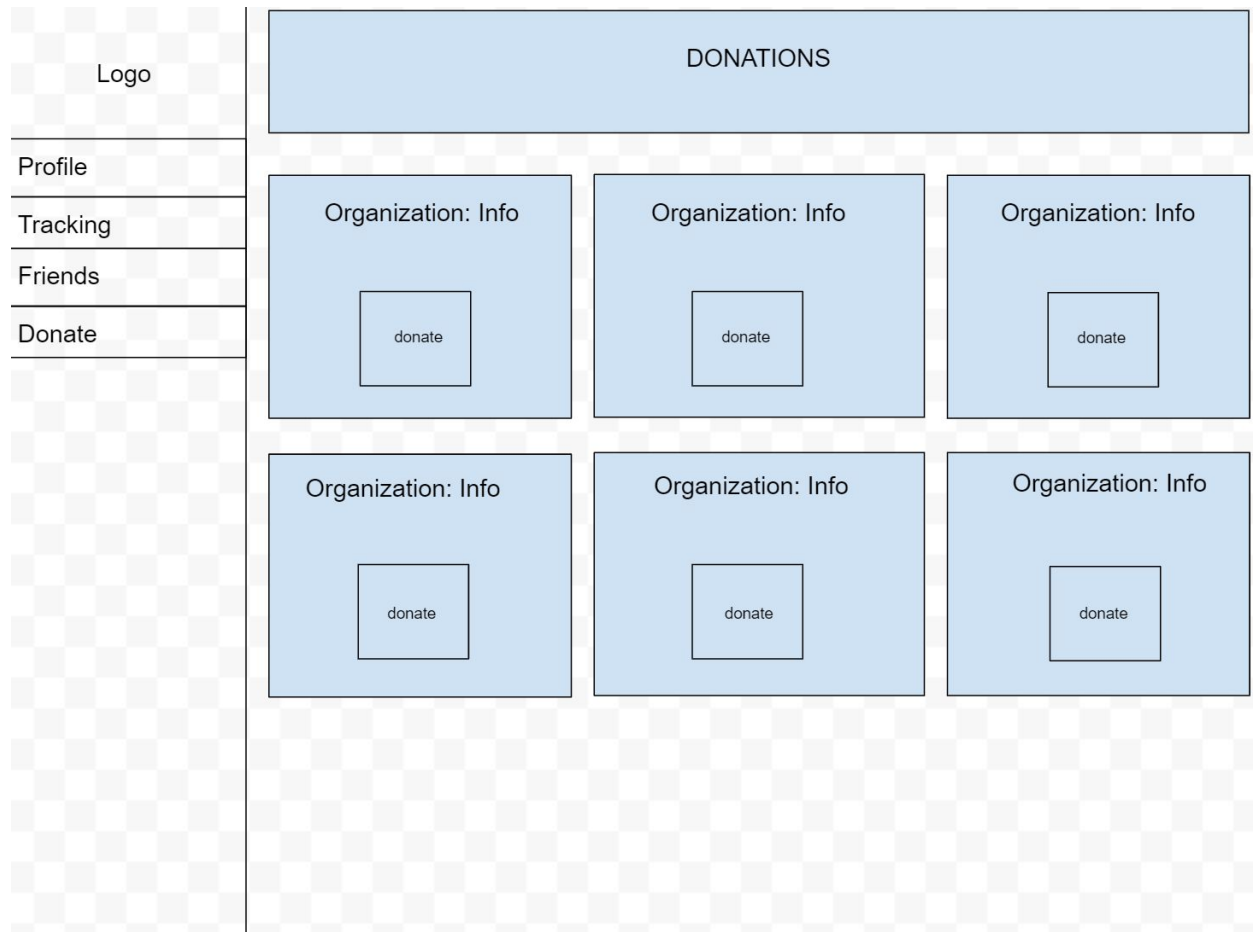
Tracking Page



Friends Page



Donate Page



Web-Service Design:

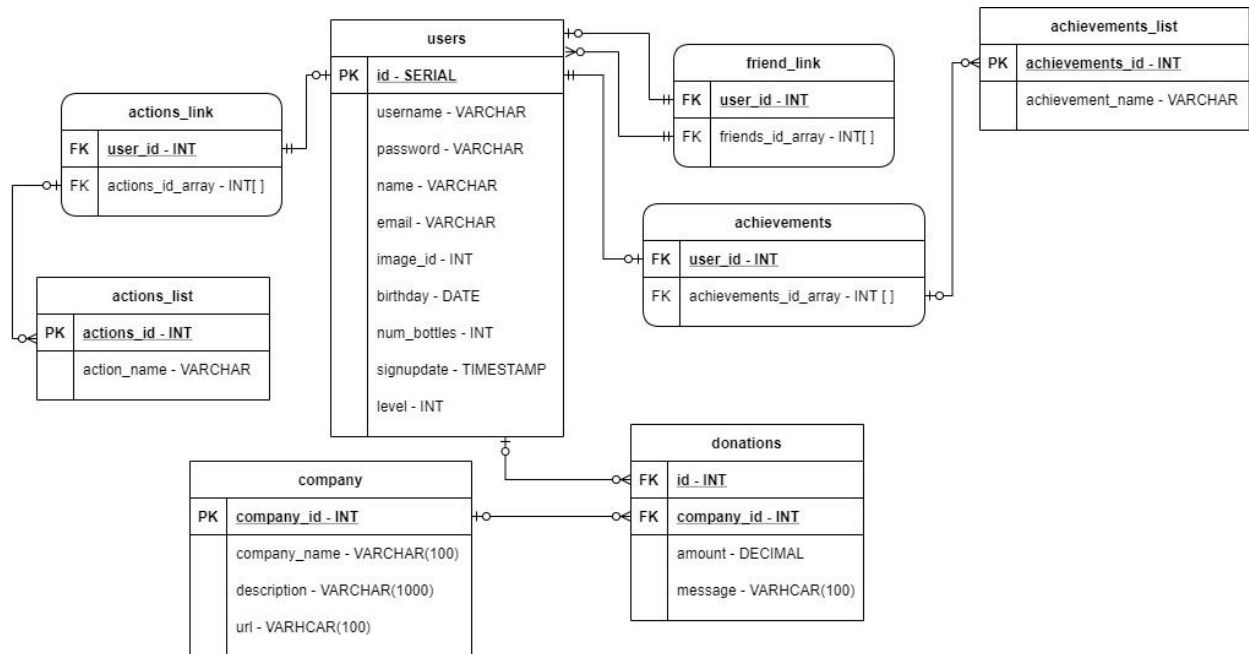
Express is used to communicate between backend and frontend: routes to get and post basic profile information, get tracking information and post newly tracked actions, get friends list and post newly added friends, get donations information and post new donations, and verify login information:

- “/” - GET: Responds with a success message
- “/signup” - POST: Accepts JSON document containing new user info. Responds with error or success and login token
- “/login” - POST: Accepts JSON document containing login information. Responds with success and login token or error
- “/users”
 - “/search?query=search_term” - GET: search user database for search term. Returns JSON array of results limit 25
 - “/:username” - GET: Responds with user info in JSON, personal info is included if the requested user is logged in. This will also accept a user id
 - “/updateInfo” - POST: Document specifying which fields to update. Backend parses and only updates relevant information.

- “/completeTask” - POST: Document specifying the task and relevant information is in the req body. The route inserts into the database and responds with success or error
- “/donate” - POST: Submit document containing payment information, donation amount, and company id. Returns success or error
 - “/companies” - GET: Returns array of companies which accept donations.
 - “/donations” - GET: Returns list of completed donations on file
 - “/top” - GET: Returns a list of the top users by total donations

Database Design:

Postgresql being used.



Individual Contributions:

Brian Mayers - Started working on backend routing and updated the database to be able to handle donations and companies. Also working on different queries to display relevant data from the database.

Link: https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/ab05ee93c8d9bcff06aa3f0496f34c903f3bd8e6

Samuel Mast - Reconfigured repository to be prepared for front end deployment. Created an app for frontend on Heroku. Was able to push the frontend onto the app so that Heroku hosts the application instead of being hosted locally.

Link: https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/d31a0b962e1dbe2f47970581cf23e652850b327f

Charlie Koepke - Worked on the tracking page code and added an “Add Task” modul form that will add a task for the user to track.

Link:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/07c48c07b1505a48d9ff4cc2bad21fd4099d97fc

Lakshya Jaishankar - Worked on restyling the home page and the login page (to have similar format as signup page). Currently, working on creating a status feed for friends page.

Link:

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/3b2aebec4a4e7fb6ab011cb57013afa3ca396648

Sarah Zendle - Created CSS stylesheet and applied it to profile page. Started working on adding functionality to friends page.

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/ff1a292dad91b9dc4e36df8dcbbf870ba479ab56

Matthew Teta - Worked on login functionality, added routes for search and friends.

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/1e84372279a46d8772201c3b2774b69ec9800e25

Rebecca Coryell - Created the Entity Relation Diagram and the Architecture Diagram. Worked on setting up a route to update the user info in the database

https://github.com/CSCI-3308-CU-Boulder/203_7_F20/commit/c152f9daedfdb9f27dc443f61d9b4d76b9728a5f

Projects / envi-203-7

Backlog

LJ CK MT BM SZ +2 | Epic ▾ Label ▾ Type ▾

▼ Tracking Page

Create the tracking page so that the user can track their environmental impact. This includes the frontend design and the backend allowing for data to be collected from user input. Date Oct 12-Oct 25
10 issues

✓ EC-58 Define track-able events	TO DO	RC
✓ EC-39 Frontend view: Dropdown metrics TRACKING VIEW	TO DO	SZ
✓ EC-56 Set up user authentication and browser cookies	TO DO	SM
✓ EC-55 Define data access rights	TO DO	MT
✓ EC-43 Create database that stores all of the user info USER DB MODEL	IDEAS	MT
✓ EC-37 Frontend view: Overall tracking page TRACKING VIEW	TO DO	LJ
✓ EC-48 Track an action TRACKING VIEW	IDEAS	MT
✓ EC-57 Define achievement and level-up structure	TO DO	CK
✓ EC-63 Upload user image capability		
✓ EC-35 Come up with metrics for measuring environmental actions TRACKING VIEW		

Quickstart

Challenges:

Donations (accepting payment)

- Backup: link other websites for donate instead of handling it through our website

Connecting between friends' pages, keeping track of different user ids

- Mitigation: keep very close track of friend information and make sure error information for the associated routes is specific and clear

Storing profile images

- Backup: present users with a selection of stock profile images, storing a profile image id instead of an entire image