

Adam Nicholas

CMPS 4928

Individual Contributions

1 Overview

Over the course of the past year, my overall contributions to Runner++ have been creating and maintaining the webpages, styling all aspects of the application, and creating additional resources to execute the look we believed would represent the school in a way the university would implement the application if they developed it. By working in tandem with Ericka, one of the 2 back-end developers on our team, I also helped to create SQL queries to push and pull data from the database and debug the PHP portions of our code. Combining both fields of contributions resulted in the best user-experience we could create, as that was our main focus considering we wanted to make the application adoptable by the school if it met its standards.

2 Front-End Development

2.1 Pages Summary

The core pages of Runner++ began as pages that both Students and Faculty would access, but we realized during testing that many pages would need to be separate to have proper functionality and explicit features for each differing user position. Simply put, the application runs off 6 main pages that follow in this order: Create Account, Login, Connect, Home, Inbox, and Logout. The user must create an account with their specified position, which they then log into. From here, depending on the position chosen, logging in will take them to a connect page that serves as a waiting room for both users while they establish a connection. Once connected, the user is given access to the home page, where a map of the campus loads, and locations are shared. On this home page, both users can access the inbox to send predetermined messages to each other about the walk. After the walk ends, both users can log out to end the session.

2.2 Styling: W3.CSS

To begin the project, I had been using default CSS to style the first developing pages. After realizing how difficult it would be to implement that for a mobile environment, I decided to switch to W3.CSS, as it was geared towards mobile development. Rather than implementing external CSS like I've done for many projects in the past, I opted to do inline CSS in order to directly control the styling of the project in a line-by-line fashion for complete accuracy and control. I found this approach easier to implement and much more followable during paired programming. To stay on theme with the school colors, I opted to use blue as the dominant color and yellow limited to a button or two. This made the app look very similar to canvas, making it

easier for users to recognize and associate with already existing resources the school uses. Due to the application being mobile-based with virtually no user using it in a desktop setting, I believe the switch to W3.CSS was a wise choice and worth the project restart in its early stages.

2.3 Styling: Font Awesome

In order to give the application a more modern feel, I added icons throughout the pages. This was my personal favorite contribution to the project because it brings more life to the app and enhances the user experience. By signing up for a free student account to Font Awesome, I was able to include a script that enabled the use of the newest icons they offer. Below are some examples of icons I implemented:

Figure 1: Font Awesome Script Code via navV2.php

```
<script src="https://kit.fontawesome.com/a6d0ff8634.js" crossorigin="anonymous"></script>
```

Figure 2: Connection, Report, and Logout Icons via connectFaculty.php

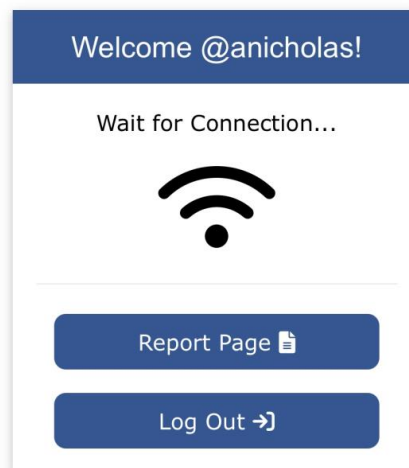






Figure 3: Profile Icons via myProfile.php

 Adam Nicholas
 anicholas
 Male
 anicholas@csub.edu

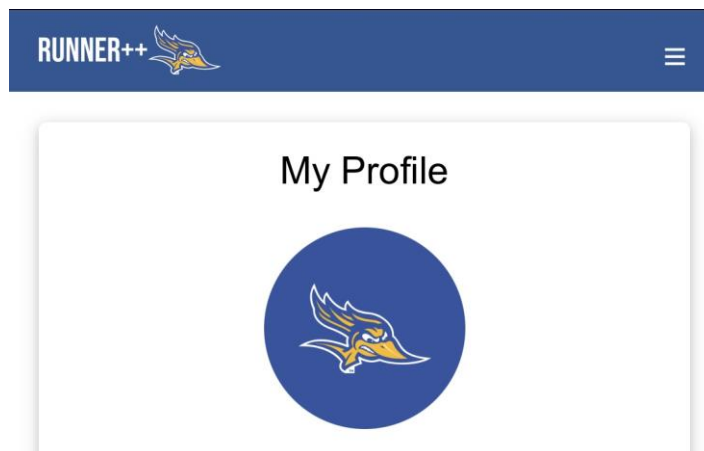
2.4 Photoshop

In order to give the application the look it needed to emulate a school-funded resource, I utilized photoshop to create the logo for our project and the user profile images. Adding these small details made the app look like it could be a resource the school currently offers, which is exactly what our group was aiming for.

Figure 4: Runner++ Logo



Figure 5: Runner++ Logo Navbar and Profile Pic via myProfile.php

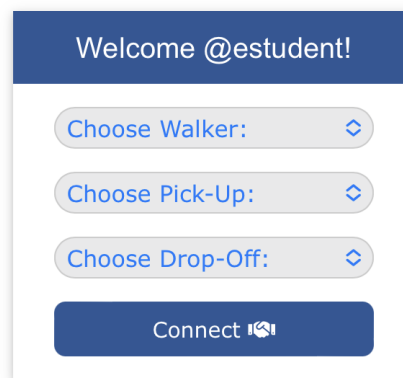


3 Student Pages

3.1 Connect

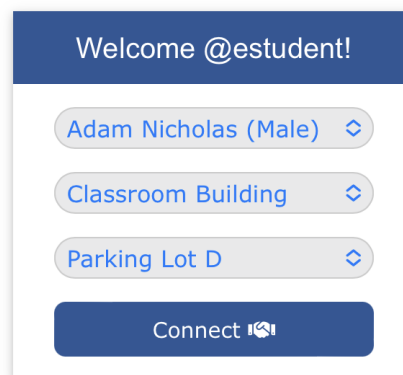
Not originally part of our starting design, with Ericka’s help, we created the student connect page as a waiting room for students and available faculty to connect while ensuring their preferred pick-up and drop-off locations are shared in case the map functionality fails during the walk. If no faculty are available when the student logs in, the “Choose Walker:” field won’t populate with any users, eliminating the possibility that a student can access the application without a connection. The pick-up and drop-off locations are existing places on campus, so users can’t enter a place that is unknown to faculty, which helps ensure the safety for both users.

Figure 6: Before Student Connect Page via connect.php



The screenshot shows a mobile interface for connecting. At the top is a dark blue header with the text "Welcome @estudent!". Below the header are three light blue dropdown menus with rounded corners. The first dropdown is labeled "Choose Walker:" and has a small blue diamond icon on the right. The second dropdown is labeled "Choose Pick-Up:" and also has a small blue diamond icon. The third dropdown is labeled "Choose Drop-Off:" and has a small blue diamond icon. At the bottom of the form is a dark blue button with the text "Connect" and a small white icon of two people.

Figure 7: After Student Connect Page via connect.php



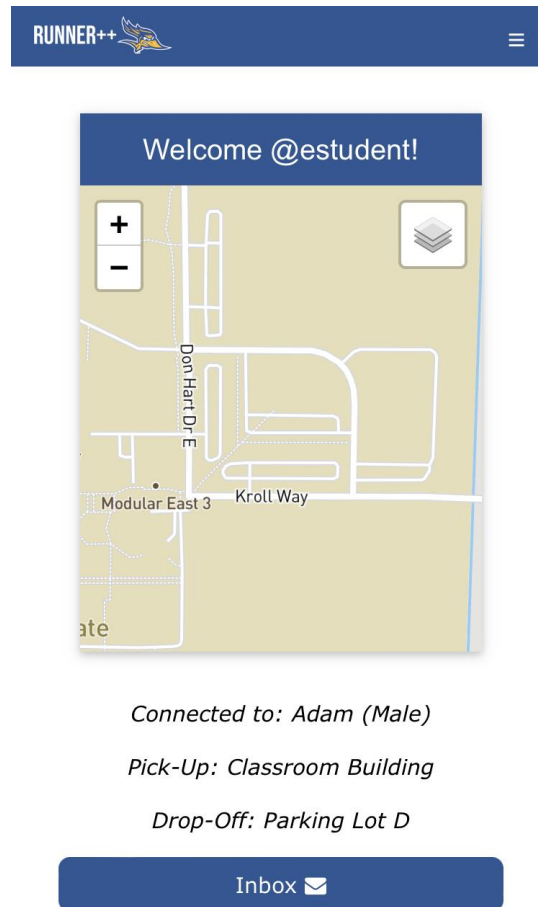
The screenshot shows the same mobile interface as Figure 6, but the dropdown menus are now populated with specific data. The first dropdown is labeled "Adam Nicholas (Male)" and has a small blue diamond icon. The second dropdown is labeled "Classroom Building" and has a small blue diamond icon. The third dropdown is labeled "Parking Lot D" and has a small blue diamond icon. The "Connect" button at the bottom remains the same.

3.2 Home

Upon connection with a faculty member, the student is taken to the home page. By dropping in the Leaflet API Daniel and Andrea implemented, I was able to style the home page

around that. We decided to have it take up as much screen space as possible and passed in who the user connected to and their desired locations in case the map failed during loading. To enhance the user experience, I added an inbox button at the bottom of the page so the user can navigate to their messages while looking at the map. Throughout the app, the navbar with a collapsible menu is included to offer navigation to other pages without taking up precious screen space. The logo in the top left corner serves as a button back to the home page at any time.

Figure 8: Student Home Page via homeV2.php



4 Faculty Pages

4.1 Connect

Similar to the connect.php page, Ericka and I added in a faculty connect page that would establish the faculty member as available to any user that logs in. While waiting for a connection, the faculty member can access previous reports with any student they've assisted. I added the Wi-Fi icon to let the user know they're waiting for a connection, and upon connecting the page

loads a new div that tells them they have made a connection. This disables report viewing but enables the faculty member to access the home page to begin the walk.

Figure 9: Before Faculty Connect Page via connectFaculty.php

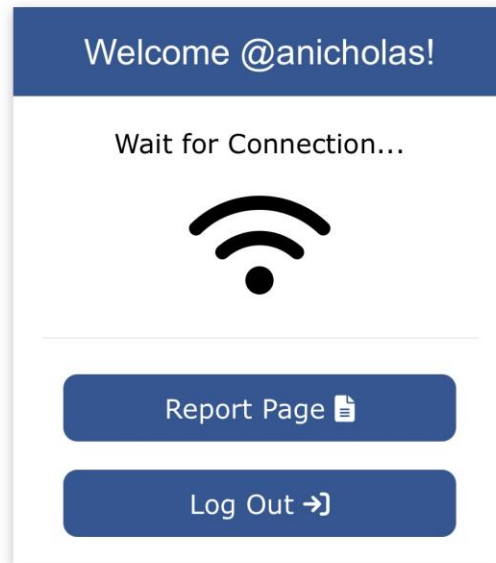
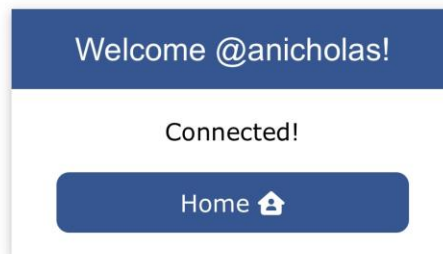


Figure 10: After Faculty Connect Page via connectFaculty.php

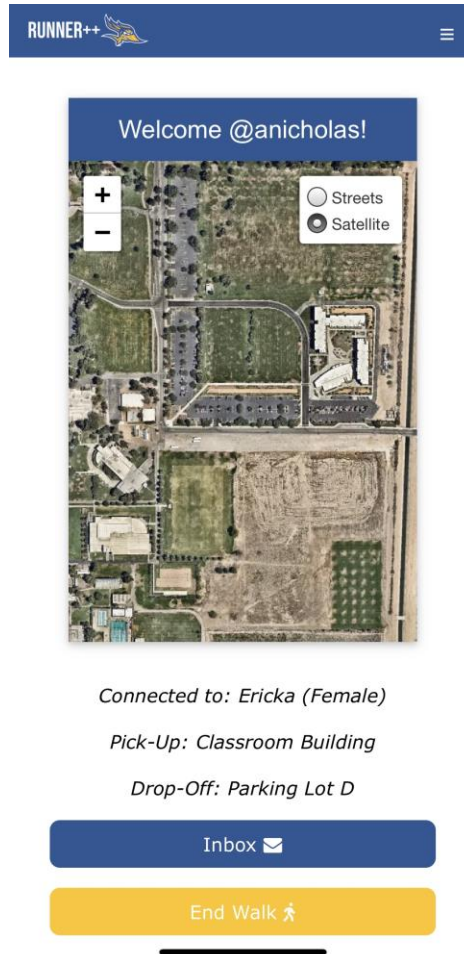


4.2 Home

Originally starting off with only homeV2.php, we decided to split the pages between positions to introduce functionality specified to the chosen position. This was done mostly for the faculty member, who has more work that needs to be done on their end since they are the user in the employee role. Different session variables are passed to each home page, so even though they look very similar they have different data being processed in the back-end. Additionally, the page has an “End Walk” button that the faculty member can press to end the

walking session. This automatically generates a report that the faculty member can access when they are rerouted to the connectFaculty.php page. From there, they can choose to log out to end their availability or reconnect with another student during their shift. The walker icon on the “End Walk” button is my personal favorite out of all the icons I used.

Figure 11: Faculty Home Page via homeFaculty.php



4.3 Report History

Only accessible to faculty via connectFaculty.php (if not connected to a student), the facultyReport.php page is able to print out any walk report from the signed-in faculty member. Each report has an auto-generated number that includes the student’s name, the pick-up and drop-off locations, and the start/end times. Each report is collapsible, so multiple reports can be viewed during the session without having to leave or reload the page. The user can then navigate back to the connect page to either reconnect or log out.

Figure 12: Walk Report History Page via facultyReport.php

Walk Report History

View Reports:

Display Report 

Report #: 288

Student: Ericka

Pick-Up: Kegley Center

Drop-Off: Icardo Center

Start Time: 2022-04-30 14:09:51

End Time: 2022-04-30 14:11:32

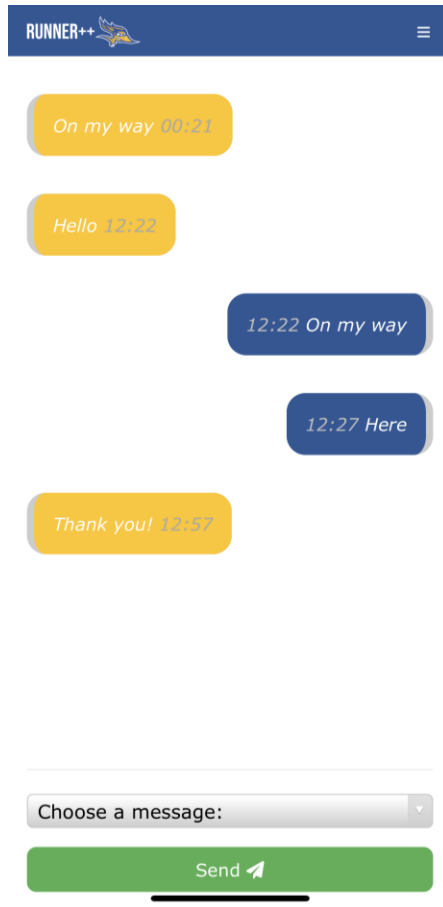
Back to Connect 

5 Internal Messaging

Working together with Ericka over many nights, I was able to emulate the way phone messages appear during messaging. By applying the styling to each individual PHP call to the database, individual messages appear on each side of the phone the way they do on text messaging conversations. Each message is marked with a timestamp from when they were sent, and autodelete the oldest message after 5 messages are sent. Using a selector to choose from predetermined messages rather than giving the user keyboard access, we kept the conversations short and simple in order to further emphasize the simplicity of the application. Each message is styled with the user's position colors, like how I did with their profile pictures in the Profile page.

- Yellow represents Student
- Blue represents Faculty

Figure 13: Inbox Page via inbox.php



6 Additional Pages

6.1 Login

For the login page (indexV2.php), I tried to make it look as simple as possible, using icons where I could eliminate words and making the app name the main focal point. On the bottom of the page, the “Create Account” and “New Password?” links navigate to their respective pages. I intentionally chose not to include the NavBar so users couldn’t access other pages without having logged in first.

Figure 14: Login Page via indexV2.php

The image shows a login form for 'RUNNER++'. At the top left is the text 'RUNNER++' in a bold, black, sans-serif font. To its right is a stylized bird logo in blue and yellow. Below the logo are two input fields: the first is labeled 'Username' and has an '@' icon to its left; the second is labeled 'Password' and has a checkmark icon to its left. Below these fields is a blue button with the text 'Log In →]' in white.

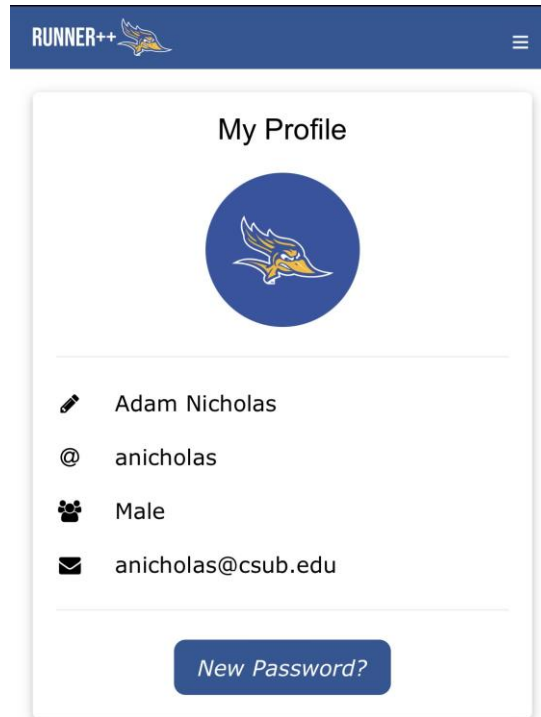
[Create Account](#)

[New Password?](#)

6.2 Profile

Using the account information given upon creating an account, I was able to display that in a profile account display commonly seen for social media. This page works for both faculty and student positions, with the blue profile pic signifying “faculty” status and yellow signifying “student” status. I also added a new password button to the page just in case users wanted to change their password, giving the page some small functionality like an account settings page would. Since we wanted the code to operate the way the school would implement it, we decided to not let other account information be changed because they would be set to the user’s school credentials, which don’t change.

Figure 15: Profile Page via myProfile.php



6.3 Create Account & Forgot Password

I applied the same styling as the login page to the createAccount.php and pwReset.php pages, adding and removing the needed fields to create accounts and change passwords. Upon submitting either page, the user gets rerouted to the login page where they can login with their new credentials.

Figure 16: Create Account Page via createAccount.php

The image shows a web form for account creation. At the top left is the logo "RUNNER++" with a blue and yellow bird icon. Below the logo are eight input fields, each with an icon to its left: a pencil for "First Name", a pencil for "Last Name", an @ symbol for "Username", an envelope for "Email", a group of people for "Gender", a checkmark for "Password", a checkmark for "Re-Enter Password", and an information icon for "Student/Faculty". The "Gender" and "Student/Faculty" fields are dropdown menus. At the bottom is a blue button labeled "Create Account".

Figure 17: Password Reset Page via pwReset.php

The image shows a web form for password reset. At the top left is the logo "RUNNER++" with a blue and yellow bird icon. Below the logo are three input fields, each with an icon to its left: an @ symbol for "Username", a checkmark for "Password", and a checkmark for "Re-Enter Password". At the bottom is a blue button labeled "Change Password->".

7 GitHub

We plan to host only our current version of the project with all of its necessary resources, as we decided as a group that we don't want recruiters or potential employers

to see our prior unfinished version. We also want to refactor the names of the files, so it is easier to read through the repository and understand what each file does by its label. I plan to use our GitHub Project repository to link the code to my personal website, as well as the video presentation we made. I'm happy with how our project turned out and I'm proud to host this on my website utilizing GitHub to have something to present during my job search as my biggest project in my portfolio.