

Sign Up

- On the sign up page enter a first name, last name, email address, and password.
- Then select how you will be using the service, as a driver or a passenger.
- Lastly, click the sign-up to complete the sign up process.

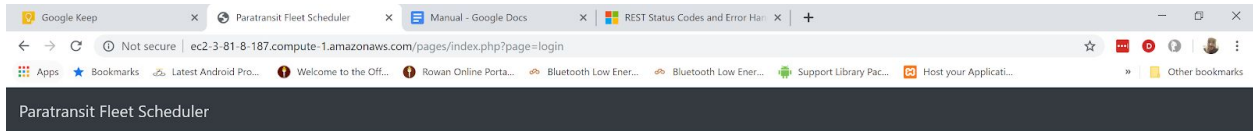
The screenshot displays a web browser window with a dark red background for the sign-up form. The form contains the following elements:

- First Name:** A text input field.
- Last Name:** A text input field.
- Email:** A text input field.
- Password:** A text input field with a visibility toggle icon.
- Service:** Radio buttons for Driver and Passenger.
- Sign Up:** A white button with the text "Sign Up".
- Already a user? Login:** A link below the sign-up button.

The browser's address bar shows the URL: `ec2-3-81-8-187.compute-1.amazonaws.com/pages/index.php?page=sign-up`. The footer of the page reads "Made with ❤️ at Rowan University".

Login

- On the login page, enter the email and password associated with your account.
- Then select the service associated with your account, driver or passenger.
- Lastly, click the login button to complete the login process.



Login

Email

Password

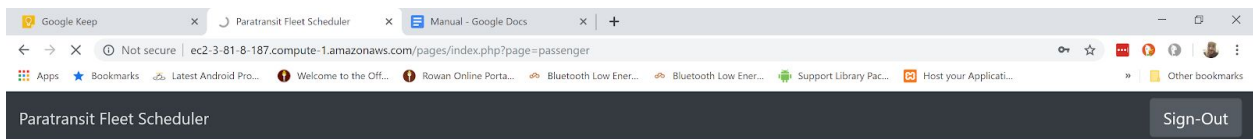
Service
 Driver Passenger

[Not a user? Sign Up](#)

Made with at Rowan University

Request Ride

- Fill in the fields for the pick up location
- Fill in the fields for the destination location
- Click the submit button, a message should appear that the ride has been successfully scheduled.



Hello, Anthony

Ride Request Form

Pick-up Address: Pick-up City: Pick-up Zip: Pick-up State:

Destination Address: Destination City: Destination Zip: Destination State:

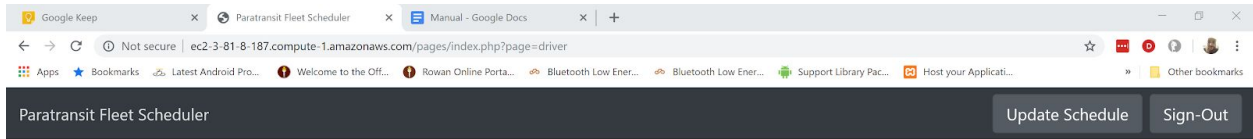
Date: Time:

Waiting for ec2-3-81-8-187.compute-1.amazonaws.com...

Made with at Rowan University

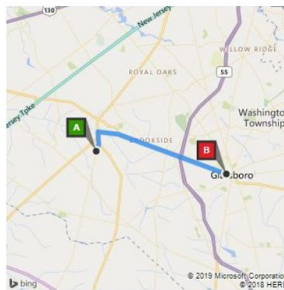
Driver View Assigned Route

- After logging in to your driver account, if you have a route assigned to you for the day, an image of your route should appear as well as a link to get directions for your route
- Click the link to open Bing Maps and get directions for your route



Hello, Anthony

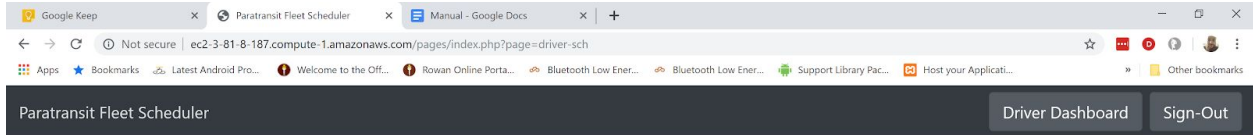
Route 1: [Click for directions](#)



Made with ❤️ at Rowan University

Driver set/update schedule

- After logging into your driver account, click the driver schedule button
- Then on the driver schedule page select the checkboxes for the days you can work and click the submit button



Your current work days: MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY

Update Driver Schedule:

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

Back-end Installation:

In order to deploy our project, the prospective client must have both a PHP and MySQL server running, and must load the empty database schema and webpages into the appropriate folders, setting access permissions as necessary. Additionally, the client must use a cron or cron-like utility to schedule the execution of the Java .jar file to run once a day, although such automation is optional as this can be performed manually; any Java runtime environment earlier than version 8.0 is not supported.

- Known problems, bugs, limitations, and unimplemented features

The most recent version of the project performs flawlessly, violating no ordering or capacity constraints. We have tested its scalability and have found that it continues to perform as expected, unless unrealistic inputs are given to it. If the proportion of ride requests to drivers becomes too high, the algorithm takes a considerable time to converge and the resultant routes are unsuitably long, sometimes exceeding a workday to drive. To compensate for this, we could have restricted user requests beyond a certain capacity for a given day's drivers, or we could have maintained a secondary table for requests to be offloaded to a third party vendor.

One of our UI limitations was the Bing Maps API's restriction of 25 waypoints in a calculated route, forcing us to split up the view into multiple subroutes. Should a client of this software wish to purchase premium access to the API, this limitation would be nonexistent and the UX would be considerably more polished. A further stretch goal for our user experience would be to introduce OAuth authentication, rather than unencrypted PHP hashing for passwords, and to include email communication with both drivers and passengers.

To detect bugs, we have tested our project on empty input, on sparse input, and on unrealistically large input. To date, we have fixed all resultant bugs, as demonstrated by the records contained in the Test Suite documentation.

Acknowledgments: Thanks to Dr. H and Dr. Wang for their assistance and guidance in developing the pathfinding algorithm.

Paratransit Fleet Scheduler Abstract

Uber and Lyft are companies in the transportation field that make obtaining a ride from point A to point B as easy as a tap of your finger. Ridesharing is also a topic of discussion with energy efficient vehicles emerging at consumer friendly prices. Combining the two and with a business demand in mind we decided to satisfy the needs of the disabled. Our paratransit fleet scheduler is created for users to request a ride ahead of time. Users will be picked up at their location dropped off at their destination in a promptly manner. The hypothesis was clear, devise a static algorithm which would account for vehicle capacity and provide riders with the shortest travel distance. Throughout the semester a few pathfinding algorithms were explored which would need to satisfy runtime, path efficiency, and capacity of a vehicle. K-means clustering proved to be a highly effective way to group requests into close proximity. Clustered groups are then supplied to the pathfinding algorithm. Early considerations for pathfinding were Dijkstra's algorithm and exhaustive search. Both proved to be cumbersome designs for the problem at hand. The final algorithm incorporates Best-First Branch and Bound pathfinding. Our dampened greedy heuristic, combined with the sorting we perform on expected value, performs considerably faster than local search alone and cuts down the explored search space by an order of magnitude in some test cases. Pathfinding scales appropriately with large input, provided the proportion of drivers grows at a similar ratio, which mirrors the real world use cases we have considered. Combining the algorithms in place along side a front-end user interface users are able to request rides with a click of the mouse.

What would you do differently if you had to do it all over again:

- Anthony Macchia:
 - Work more on the back-end of the project
 - Do more work in the beginning of the semester
- Raymond Pickett
 - If i were to do this project again, I would start writing programs and using a database sooner.
- Tyler Rambo
 - Gain more insight on all of the various parts of the project, especially the coding of the algorithm.
 - Try to coordinate more scheduled meetings during each week to keep on topic. Schedules permitting.
- Wes Holman
 - Within the first two weeks of the project, I would have built a brute-force solution to the problem to improve upon, if I had to do it over again
- Kieran Walsh
 - For the back-end I would have made more wrapper classes to hold information, instead of having all the data stored in Arrays. With all the multi-dimensional arrays it was harder to find all the stored information.
 - Considering I didn't know anything about databases, it would have been useful to study them more, as it would have helped future programming.

What you learned from working in a group:

- Anthony Macchia:
 - I learned that communication is a key aspect to the success of a project.
 - It is best allocate work based on where people's strengths are
- Raymond Pickett
 - I learned to stay up-to-date with what the group is doing, have regular group meetings, and actively help complete whatever part of of the project is in progress at the time.
 - I learned about using databases, github, and slack. I learned about K-means clustering and ridesharing algorithms.
- Tyler Rambo
 - Everyone has the tendency to lose focus due to other courses and structured meetings provided an effective way to regain control.
 - Allowed me to practice explaining functionalities of my modules to other group members who may not initially understand.
 - Learned how valuable version control tools like GitHub can be working in a large group.
- Wes Holman

- I learned how to delegate responsibilities fairly and work with partners to accomplish subgoals
- Kieran Walsh
 - I learned a tremendous amount of databases, and it helped me in other classes as well. Before this class I wouldn't know how to create a table, Insert values, or Select values from the database and now I can do it very easily.
 - I learned a lot about GitHub. I have used it before, but didn't really understand how useful it was. The merging of code was extremely helpful and easy.
 - I learned a lot about programming in general. Having to work on a big project like this helped me hone my skills as a software developer.