

tigerride™ Product Guide

For Users

What is tigerride?

Tigerride is a student-run application created to streamline the process of students sharing transportation services to and from the Princeton University campus. Sharing such services allows users to save money, travel more safely, as well as take a step towards a greener environment. Users fill out a simple form to find other users to share rides with, and have the flexibility of requesting to join and leave rides at any time.

User Guide

Note: All users of tigerride must be Princeton students or faculty members with a valid netID and password because our website utilizes a Central Authentication Service system to make the app safe and reliable for all users.

Logging In

1. Navigate to <http://tigerride.herokuapp.com/>.
2. Click the “login” button at the bottom, which takes the user to the Princeton University Central Authentication Service authentication page.
3. Fill out input fields for NetID and Password, click “LOGIN” and proceed with preferred method of DUO security authentication.

Creating a Profile

1. The first-time user must create a profile in order to create or view rides.
2. Users are presented with two options: creating a new profile from scratch, or importing their information from Tigerbook.
 - a. Starting from scratch:
 - i. Fill out fields for preferred first name and last name.
 - ii. Input phone number.
 - iii. Upload a profile photo.
 - b. Using Tigerbook information:
 - i. clicking the button automatically imports the user’s name, netID, and photo.
3. Users can update their profile information anytime by navigating to their profile page by selecting the “profile” option on the menu bar and clicking the “edit” button.

Finding Rideshare Buddies

1. Once the user logs in, they are redirected to a welcome page with presented with two buttons. Click on “plan a trip.”
2. Under “origin,” select desired origin from the dropdown menu. Options are EWR, PHL, JFK airports and Princeton University.
3. Under “destination,” Select desired destination from the dropdown menu. Options are EWR, PHL, JFK airports and Princeton University.
4. Under “date,” input the day, month, and year of departure. Clicking the small arrow on the right opens up a calendar for easy selection.
5. Under “time frame,” there are two fields. In the first field, enter the earliest possible time of departure. In the second field (after “~”), enter the latest possible time of departure
 - a. Tip: Entering a flexible time frame is more likely to return more options.
6. Under “notes,” enter any information that may be useful for potential ride sharers to know. Some things to consider:
 - a. desired rideshare option (ex. Standard Lyft, Lyft XL, Lyft Lux, Uber X)
 - b. flight number (be cautious!)
 - c. terminal details
 - d. amount of luggage
 - e. preferred seating arrangement (ex. next to the driver, in the back)
 - f. desired number of rideshare partners
 - g. your airline is known to be frequently delayed
7. Click “submit.”
8. If user is the first to submit a ride in their desired time frame, they are redirected to a page that displays the following message: “Your ride has been created and we will notify you via your Princeton email when other students join your group!”
 - a. User will receive e-mail notifications of trip details and group members when others join their group.
9. If another user has joined the ride immediately after the current user has submitted his/her ride, the following message will be displayed: “A rider beat you to it and has already joined your ride!” The user can then see this ride group by clicking on “my rides,” which navigates to his/her ride history.
10. Otherwise, user is redirected to page displaying all open rides that match their search. User can join a ride. This notifies all members in the group via their Princeton e-mail of the trip details.
11. If the user does not join someone else’s group, tigerride automatically creates a ride with their own preferences for others to join in the future.

Viewing Your Ride History

1. From the home page (the first page user sees after logging in, can be accessed by clicking “home” in menu bar), click “see all rides.”
2. Under “current rides,” you can view all your current open rides. Under “past rides,” you can see all your completed rides.

Re-Searching for Rideshare Buddies

1. Note: Re-searching for ride share buddies is only possible if you have not yet joined a group.
2. From the home page (the first page user sees after logging in, can be accessed by clicking “home” in menu bar), click “see all rides.”
3. Under “current rides,” locate the ride you would like to find rideshare buddies for and click “see more.” You will be taken to a page that shows all the details of the ride with the option to “continue search.”
4. If there are no current rides that match their ride details, the student will be redirected to a page that displays the following message: “Your ride has been created and we will notify you via your Princeton email when other students join your group!”
 - a. User will receive e-mail notifications when others request to join their ride.
5. Otherwise, user is redirected to page displaying all open rides that match their search. User can join a ride. This notifies all members in the group via their Princeton e-mail of the trip details.

Deleting a Ride

1. Note: Deleting a ride is only possible if you have not joined a group and are the initial creator of the ride, and completely removes the ride from the tigerride database.
2. From the home page (the first page user sees after logging in, can be accessed by clicking “home” in menu bar), click “see all rides.”
3. Under “current rides,” locate the ride you wish to delete and click “see more.” You will be taken to a page that shows all the details of the ride with the option to click “delete ride.”
4. You will be taken to a confirmation page that asks “are you sure you want to delete this ride?” Click “delete!” Note: this action cannot be reversed.
5. The ride is deleted from your ride history as well as the database.

Completing a Ride

1. Note: Completing a ride is only possible if you have joined a group.
2. From the home page (the first page user sees after logging in, can be accessed by clicking “home” in menu bar), click “see all rides.”
3. Under “current rides,” locate the ride you wish to complete and click “see more.” You will be taken to a page that shows all the details of the ride with the option to click “complete ride.”
4. You will be taken to a confirmation page that asks “are you sure you want to complete this ride?” Click “complete!” Note: this action cannot be reversed.
5. The ride is now moved down to the “past rides” section and other users can no longer request to join your ride.

Leaving a Ride

1. Note: Leaving a ride is only possible if you have joined a group.
2. From the home page (the first page user sees after logging in, can be accessed by clicking “home” in menu bar), click “see all rides.”
3. Under “current rides,” locate the ride you wish to leave and click “see more.” You will be taken to a page that shows all the details of the ride with the option to click “leave ride.”
4. You will be taken to a confirmation page that asks “are you sure you want to leave this ride?” Click “leave!” Note: this action cannot be reversed.
5. You are no longer a part of this group, but your individual ride will still be visible under “current rides” in your ride history.

Rating a Rider

1. Note: You can only rate a rider if you have completed a ride with him/her.
2. From the home page (the first page user sees after logging in, can be accessed by clicking “home” in menu bar or the “tigerride” logo), click “see all rides.”
3. Click on any rider’s name in your past rides. This will take you to their user profile, where you will see an option to click “rate.”
4. You will be taken to a page that will allow you to rate them on a scale of 1 to 5 (1 being the worst and 5 being the best).
5. Upon success, you will be taken to a page that allows you to navigate to your ride history once again by clicking the button “see my rides.”

User Profiles

1. You can view a user profile at any point where their name and a button to their profile shows up. If they do not have a user profile, you will be taken to a page with a link to their TigerBook page.
2. Each user profile will always display the first name, last name, rating, netid, and number of trips taken. However, when you are currently in a ride-share group with other users, their profiles will also display their phone numbers for ease of contacting them. You are also able to view the phone numbers of each user you have a past ride with.
3. As explained earlier, you are also able to rate a user in their profile (but this feature is only available if you have completed a ride with him/her).

Addressing Questions, Concerns, and Suggestions

1. Click the “contact” button on the menu bar.
2. Email any of the four developers listed on the page with your question, concern, or suggestion.

Logging Out

1. Click the “logout” button on the top right corner of any page.

For Developers

How it Works/Implementation

We used Python 3.7 on Django 2.2. For the purposes of describing project organization, we'll mention specific folder names. Our Django project is called “tr,” abbreviated for tigerride, and is divided into two Django apps, named “create_ride” and “login”. We used the two apps to manipulate our two different types of Django models: one for rides that users can create, and one for information about users, containing fields such as their name, netid, phone number, rating, and a link to their photo, optionally served through Tigerbook. Each app primarily contains a models.py for the app's model, as well as a urls.py and a views.py for isolated use cases. We recognize that we can consolidate our apps, but we decided to separate the two for modularity and migrations/database debugging.

Upon clicking a button or being redirected to another page, the usual Django scheme of routing from hooks in a urls.py file (included in the directories “tr, create_ride, and login”) to a function in {{directory}}/views.py is in place. In particular, we use the path() function in {{directory}}/urls.py, instead of the outdated and soon to be deprecated url() or re_path(). This, of course, relies on using an up-to-date version of Django.

We connected to our Postgres database in tr/settings.py by 1) initially connecting to an sqlite3 database as a default/backup, then 2) resetting our database connection variables by importing the Python package dj_database_url and setting our database connection variables to track the free-tier Postgres. When debugging on localhost, be sure to comment out the three lines starting with “import dj_database_url”, and be sure to uncomment before deploying to Heroku.

Each ride, complete with all its information, is stored in the Postgres model InputRideInfos. To organize these rides into groups, we gave each group a unique identification number -- whenever a user joined a ride, we simply updated his/her entry in the ride database to match the identification number in a group, and vice versa when a user wanted to leave a ride.

When a user submits a ride schedule through tigerride, the searchResults() function in tr/views.py is called. This function queries the database for existing rides and returns a list of rides the user might be interested in. Using the Django QuerySet interface, we render all the rides that 1) are open, 2) match origin and destination to the submitted ride, 3) match date, and 4) overlap in times, that is, have start times before the

submitted ride's end time and end times after the submitted ride's start time. We considered adding an arbitrary range value (for example, one hour) to match start times more closely, but we decided on this looser filter to show a broader selection of rides to give users more options, especially at the beginning of marketing for tigerride when there may not be a large volume of users. In order to grab the submitted ride from the user for filtering, we pass a unique ride_id as a GET parameter. Additionally, we use ride_id to exclude the user's submitted ride from the list of search results. A known bug for search results is that rides spanning multiple dates (from 11pm to 2am, for example) may not match correctly, depending on the user input. We decided that the issue was not currently pressing, but we will be implementing a fix for this bug before the app is advertised for use.

To consider the cases in which a user never joins a group or obtains a ride-share group, we decided to implement an "expiry" feature -- if the ride date is over one day prior to the current date, then we will automatically update and close the ride, placing it under the past rides.

We implemented a user profile page to view information about each user (such as name, photo, and rating) prior to joining a ride. Similar to before with searching for matching rides, we grab and render a user's info (QuerySet) by netid passed in by POST request from buttons in a rendered ride.

We give users the option to rate other tigerride users through a button on their user profile. This leads to an HTML form which allows users to rate with either 1, 2, 3, 4, or 5 out of 5. When this form is submitted, the rateRider() function in views.py executes, which grabs the old rating and number of times the user has been rated, multiplies them and adds the submitted rating, divides by the new number of times the user has been rated, and updates the rating and count values in the database. For a new user, ratings default to 5 and rate count defaults to 0. We decided these values because the 5 is visually appealing for new users, but a count default of 0 by our algorithm means that a user's rating will only be influenced by other users instead of other users *and* a great score of 5.

One potential flaw in our ratings system is that we impose very few limits on who can rate a user, or how many times they can rate them, e.g. the only limit is that you can't rate yourself. This potentially means that someone can troll a user by rating them as a 1 twenty times in a row, or that someone's friends can inflate a user with a bunch of 5s. We decided this was not a significant issue, because our open interface for creating and joining rides means that a determined user can completely control their rating in almost any scenario. The fact is that any kind of ratings feature assumes a certain level of trust in its users, and we saw it as outside the scope of our project and our mission to regulate our users in this manner.

Software Systems and Resources Used

We are hosting tigerride on Heroku, and we're indebted to them for providing a free-tier usage that can serve the audience for our app. In addition to working on Django, we use the `django-heroku`, `dj-database-url`, and `psycopg2` packages to connect to Heroku and our Postgres database. We also use Lance Goodridge's `uniauth` package for connecting with CAS, which has saved us a lot of time in the long run. We also used the Tigerbook API to serve the names and images of users who choose to connect with Tigerbook, and we interact with that API using Kenneth Reitz's `requests` library. Princeton's Office of Information Technology authenticated the URL `tigerride.herokuapp.com` for CAS access for the duration of this semester. Of course, many thanks to StackOverflow.com, Django documentation, and YouTube tutorials