Kyle Lim, Jaehong Lee, Jonah Chesnut

CPTs 489 Project Team & Project Proposal

1. Project Overview

The trucking industry is the backbone of commerce, facilitating the transportation of goods over vast distances. To address archaic methods of communication between truckers and brokers in the trucking industry, Truckstop is a platform to facilitate communications between these two parties; for their services, they charge a non-trivial premium. However, their premium is currently justified – Truckstop has a large network of truckers and brokers.

This project will take inspiration from Truckstop, the leading company acting as liaison between truckers and brokers, while addressing key areas where modern solutions fall short. For instance, real time tracking, automated matching algorithms, and transparent rate insights should be standard to foster a fair and efficient marketplace.

This project will empower independent truckers to find loads efficiently and eliminate the anti-consumer paywall for standard modern solutions. Simultaneously, it will provide brokers with enhanced tools to connect with a broader network of reliable truckers, streamline their operations, and reduce operational inefficiencies. The result determined by the success of this project is an end-to-end democratized logistics system.

2. Technologies:

Now, since this is a project for a class where it has a basic MVP prototype, this project would use the following technologies:

- HTML, CSS, JavaScript
- DGraph
- PostgreSQL

The combination of the above technologies creates a cohesive tech stack for building a modern, scalable, and efficient web application. Additionally, they have permissive open-source licensing.

2.2.1 Graph Database:

DGraph will be the primary broker facilitator to efficiently find truckers. reduce trucker idle time. Each registered broker will exist as a node in the graph as well as each trucker. The following is the trucker node schema:

truck_node:

- Id (Unique identifier): Ensures each trucker is uniquely identifiable.
- Name (String): Trucker's name.
- Capacity (Integer): Storage space in truck.
- Status (String): En route, idle, unavailable.
- Rating (Float): Community rating.
- Review (String): Reviews on a trucker.
- Current city or destination (city_node): Edge associated has destination time if destination.
- Qualifications (String): Hazmat, tanker endorsement, etc.

city_node:

- Id (Unique identifier): Ensures each city is uniquely identifiable.
- Name (String): City name.
- Trucks (truck_node): Trucks en route and idle trucks.

brokerage_node:

- Name (String): Company name.
- Brokers (broker_node): Employees of brokerage.
- City (city_node): Business location; can traverse city node to get state.
- Rating (Float): Aggregation of broker node ratings.
- Review (String): Reviews on a brokerage.

broker_node:

- Name (String): Broker name
- Truckers (trucker_node): Truckers working for broker.
- Company (brokerage_node): Can traverse node to get company data.
- Rating (Float): Brokers individual community rating.
- Review (String): Reviews on a broker.

With the relationships modeled in this graph database, several relationships can be modeled with this schema. If a broker is deciding whether to accept a load for a specific city, brokers can search that city on the load board page of the site. If truckers are en route to that location, or in that city, the brokerage can decide to accept the load.

In the graph the query begins at the city, then this query will traverse all truckers attached to the city. From here the query gets information from current city or destination attribute, rating, capacity, and qualifications. The broker can then message the trucker to ask whether they can accept the load. The trucker then views the company directly from the broker's message.

If the trucker clicks on the broker who messaged them, the broker is associated with a rating and their overall company rating. If the trucker agrees the trucker will accept the load, and the broker_node will attach to the trucker node.

If the trucker was en route to their location when they accepted, then the trucker will immediately have business by the time they reach their destination. This reduces idle time for truckers and increases the availability of truckers for brokers. There are several other relationships that can be modeled with this schema, but for brevity, those relationships will be omitted.

2.2.2 PostgreSQL

PostgreSQL will be used for transactional data that does not fit well into the graph database. For example, billing, storing credentials, role-based access, records of load bookings, and for document storage. Since this is an MVP model, our team will focus on storing credentials and role-based access.

3. Expected Pages:

While our overview of the project concept and technologies has laid out the function of our proposal, this section will present a tentative plan for what pages we will need for our project. We are guessing 8 pages will be needed for our webpage project.

- Home Page: A basic overview for new visitors of the website, giving a simple overview of what the site does and having buttons to sign in and traverse the rest of the app.
- Login: A page that allows a user to choose to login as a broker or trucker, which leads to selected roles respective page after confirming their identity.
- Registration: A page for a user to sign up as a broker or trucker.
- Trucker Overview: A page for logged in truckers to see their rating and stats, update information such as their location or truck capacity, and view information about past, current, and potential future jobs.
- Broker Overview: A page for logged in brokers to see their stats, as well as get information on their current hired truckers, and any truckers available for work in their area.
- Brokerage Overview: A page for logged in brokers who have been granted brokerage access to manage their individual brokers, as well as see those brokers' statistics.
- Job Information: An information page displaying in-depth information about a user selected from an overview page, as well as a proposed job, allowing truckers to decide whether to take a broker's job, and allowing brokers to decide whether to hire a trucker for a job.
- Rating/Review: A page for logged in users to give ratings or reviews once they click on profiles of registered truckers, brokers, brokerages.

4. Team Members:

- Jonah Chesnut https://github.com/JonahChesnut
- Jaehong Lee https://github.com/Jaehong-username
- Kyle Lim https://github.com/hiefenhoomer

Our team for this project is composed of three undergraduate computer science students from Washington State University. The three of us see this project as a great way to learn about full-stack web development, and hope to finish this project with an increased understanding of both the organization and technical skills required to make a polished and functional website.

5. GitHub:

Our team will store our project files on a GitHub repository to facilitate the organization of our project as well as increase our ability to work as a team. The link for our repository is below:

https://github.com/Jaehong-username/cpts489-web-project.git