**DataTree Project: Keeping Your Data Safe, Secure, and Stylish**

Ian Siple

Jared Endicott

Alexander Wade

Huy Hoang

Ali Chowdhury

CSC 4110: Software Engineering

Professor Myers

3/7/2023

Mission statement:

The big issue that every company faces is data storage. Storing data has become one of most common yet complex tasks to achieve for any company. Luckily, with DataTree, your common issue is not a complex one. At DataTree, we store your data with ease. We make your data like a tree so you can leaf the rest to us. Your data is our utmost priority, and we stop at nothing to make sure your data stays to its roots.

Timeline

Starting from humble beginnings, us five had our work cut out for us. We started off with seeing who works best with what. We found that Ian works best on the backend, he coded a large portion of the overall backend code. Alex had the front end down pat and established a GUI for us to use. Jared was responsible for the website where you can download our service. Huy and I were responsible for writing, I on the documentation side and he on the ToS side. Once we first got into the swing of things, we got right into work. Alex and Ian made great strides on their respective ends, two sides of the same coin working together like birds of a feather, creating our application that you now see before your very eyes. Jared and Huy got work done on the website and coding compliance, respectively. I got our documentation down, our slides and testing.

Issues

Our first issue: The website: We needed a way to push our service to anyone that has access to the internet. We started with the triple threat basics: HTML, CSS, JavaScript, then we found some Bootstrap templates to manipulate and use for our needs. Once we did that, all that was left was to add our download links and write in our mission statement. All in all, the website proved to be a bit of a challenge to push through, but with a little bit of water, a little bit of sunlight and a whole load of photosynthesis, we grew to great heights.

Our second issue: Manipulating the data. Our first task was to get values from a csv file, or comma separated value and append to the database. This helps the user take any data they have and create their own way to store and search through the set. This was definitely not a job that was done overnight, and this was most definitely a job that had to have multiple partitions to split amongst the group. Ian was our main man for the job. Ian created the majority of the functions that the program uses and was a driving force to our commitment to our clients. There are other aspects that our group handled in regards to programming our DataTree application. The GUI, for instance, is what Alex worked on and was able to create an interface where the user can easily and visually see his or her dataset being changed before their very eyes. This approach and our overall approaches to how our data is handled makes it very easy for the end user to use our powerful methods of data manipulation and searches and it serves as a beacon of what our customers can look forward to when growing with us.

There were other minor issues that the rest of the team tackled. In our weekly coding reviews, we were able to discuss current and future plans that our team was able to hack down. One of the bigger issues that we worked together on as a team is how we pull the data from csv. Using the methods we discussed in class, we created an algorithm that took input from a given csv file to store it in our database. We all took part in the initial step and it was a combination of 3 people in the group that created the first initial portion of the project which kicked off more things to come. One of the other wrinkled flags that we ironed out was a major issue one of the csv files had: the orderID number was not aligned. One of the sample data sets had an inconsistent numbering for the file. We created an algorithm to fix that issue so that our data was more consistent in the procedural sense. This just shows that even though our team has their own issues, we can still come together and solve a greater problem for the collective.

The next issue we tackled was if our code was compliant to PEP-8 standards. This was an issue that was easily fixed by Huy, who also wrote up our Terms of Service. We used an extension called PyLint which made sure to give us a high hard time if we didn't comply with programming standards. It was very necessary as we believe a strict environment on the development side leads to a structured one on the production side. The terms of service was also a major issue to tackle. We made sure that every use case for harm does not hold us liable.

So, why should we be chosen? I think it’s very clear. We have a structured way of developing that gives you a flexible way of life. Your data storage issues are a thing of the past once we’re under your belt. Not some of the guesswork is taken off your plate, we make sure that all of your plate is cleaned, not a singular dirty dish left behind. At DataTree, data is the most important aspect of our service. I mean, why wouldn’t it be? We take the extra step in making sure that your data is protected and secured. Our team of elite programmers make sure that once your order is processed, it is confined and locked without a shred of sunlight to beckon for.

So what’s next?

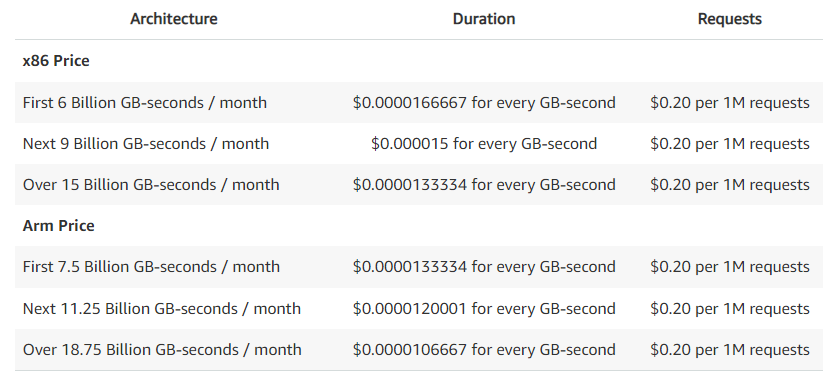
Our next step at DataTree is to grow our tree all the way up to the clouds. That’s right. Your data will be safe and secure when it’s getting its hydration from our new proprietary cloud server. This way, your data will be protected and secure.

Pricing

So the plan that spoke out the most to us was the AWS Lambda Pricing from Amazon. The pricing for AWS Lambda is based on the amount of memory allocated to the function, the duration of the function, and the number of requests made to the function. The cost of using AWS Lambda can be calculated based on these factors, and is subject to different pricing tiers.

For example, using the pricing table for x86 architecture in the US East (Ohio) region, the cost for executing a function 1 million times per month, with each function taking 500 ms to execute, and with 512 MB of memory allocated to the function, would be approximately $5.55 per month. This estimate is based on the assumption that the function falls within the first pricing tier, with up to 6 billion GB-seconds per month in usage.

However, it's important to note that this is just a rough estimate, and the actual cost may vary based on several factors, such as the region, the number of requests, the duration of the function, and the amount of memory allocated. Therefore, it's crucial to carefully evaluate the cost implications of using AWS Lambda before making a final decision.



Plans

At DataTree, we offer a flexible pricing model to fit your unique needs. Our pricing plans are based on the amount of data storage you require and the level of support you need. We offer three pricing tiers: Basic, Pro, and Enterprise.

Our Basic plan starts at $9.99 per month and includes up to 10GB of data storage, email support, and access to our basic features. Our Pro plan starts at $19.99 per month and includes up to 100GB of data storage, priority email support, access to our advanced features, and a 30-day money-back guarantee.

For larger companies with more complex needs, our Enterprise plan offers custom pricing based on your specific requirements. With the Enterprise plan, you will have access to all of our features and the highest level of support. Our team will work with you to create a personalized plan that meets all of your data storage needs.

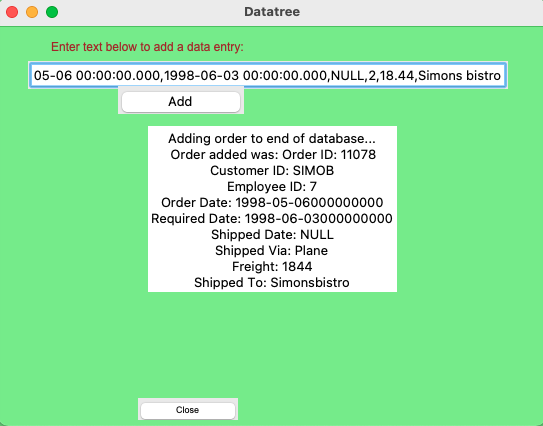
We understand that every business is unique, and we strive to provide a pricing model that is both affordable and scalable. With DataTree, you can be sure that you are getting the best value for your money. Contact us today to learn more about our pricing plans and how we can help your business grow.

| Meeting # | Topics | Details |
| --- | --- | --- |
| 1: 2/7/2023 | Separation of duties | Discussed who gets which part  -Front End: Jared and Huy  -Application: Ian, Alex, Ali  -CC Huy  -Written: Ali  Decided on naming convention (snake\_case) |
| 2: 2/9/2023 | 1st code review | Discussing our first code push, squashed issues with accessing branches  Discussed how our front end should look |
| 2/14/2023 | No formal meeting | Crew worked on code for each aspect |
| 2/16/2023 | Restructured how each person works on each project | Refactored code |
| 2/21/2023 | Finished product | Implemented GUI, Tested program |
| 3/5/2023 | Pricing | Found out pricing for our application |

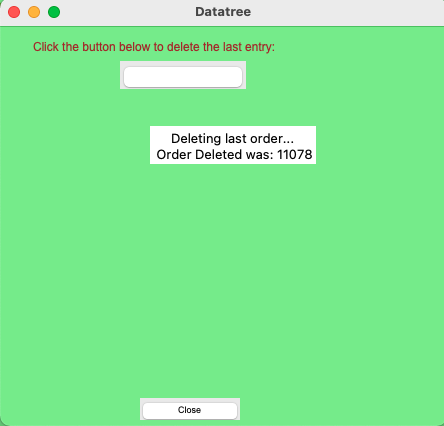
| Progress | Week | Details |
| --- | --- | --- |
| 2/09/2023 | 1 | Pushed prototype code to the repo |
| 2/11/2023 | 2 | -Fixed a big issue with order-details.csv  - Basic frontend work  -Setup Github repo  -Come up with basic strategies on how we plan to search through and filter data.  -Finish forestview\_datastructure class  -Tkinter discovery |
| 2/14/2023 | 2 | -Finished internal code  -started on GUI script  -Basic webpage layout complete  -Started Google Slides |
| 2/17/2023 | 2 | -Finished GUI code  -Polished search function  -Webpage complete  -Finishing Doc |
| 3/5/2023 | 4 | -Created new GUI  - Finished presentation |

Technical documentation:

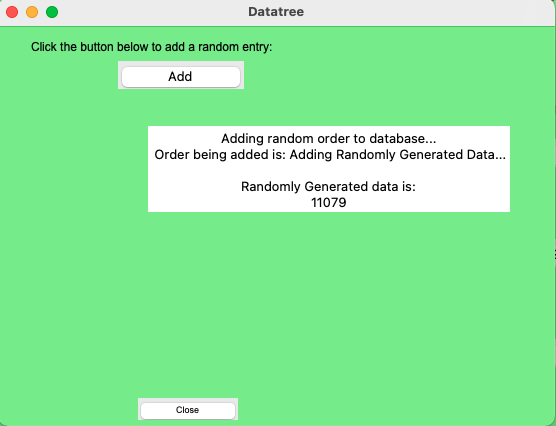
Adding an order:



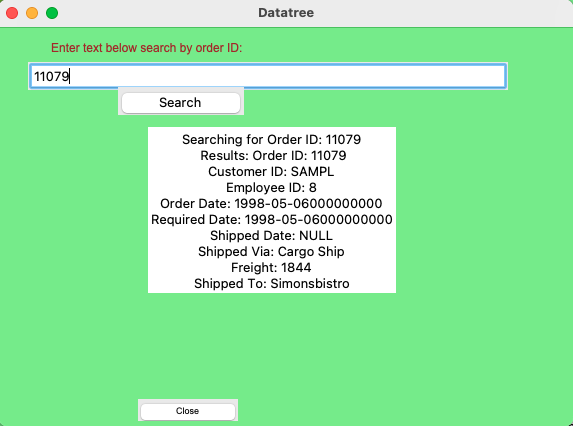
Deleting last order:



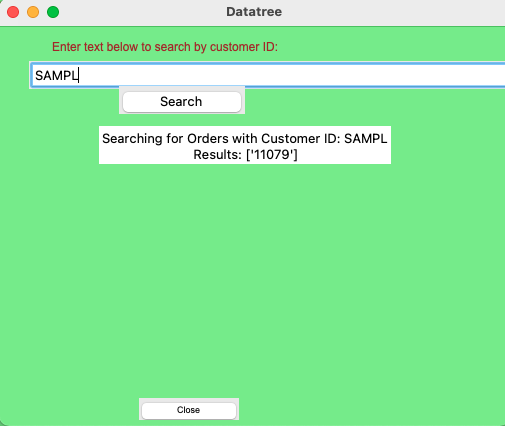
Adding random order:



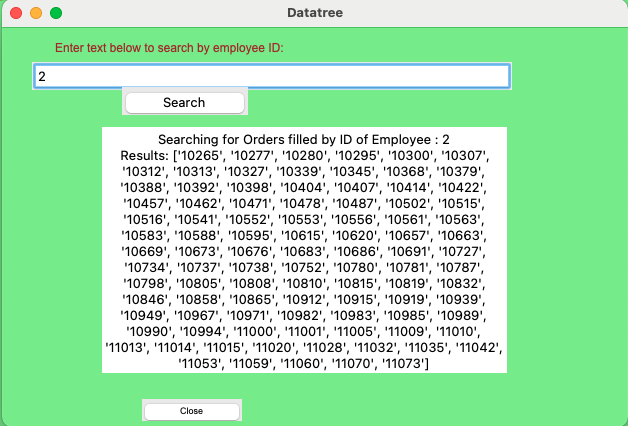
Searching order:

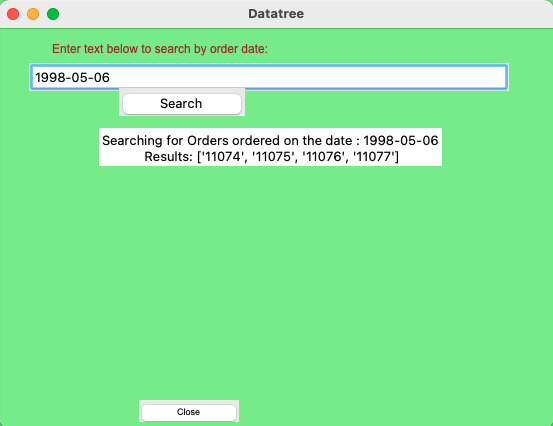


Searching for customer ID for all SAMPL data:

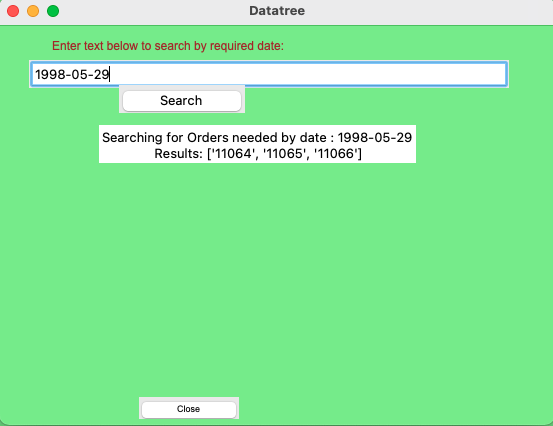


Searching for orders done by same employee ID:

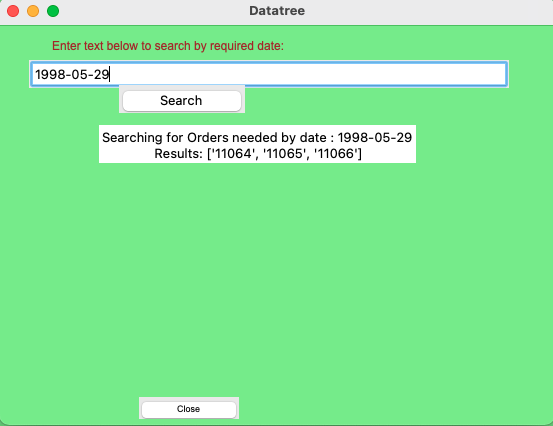


Searching for orders ordered on date:  
 

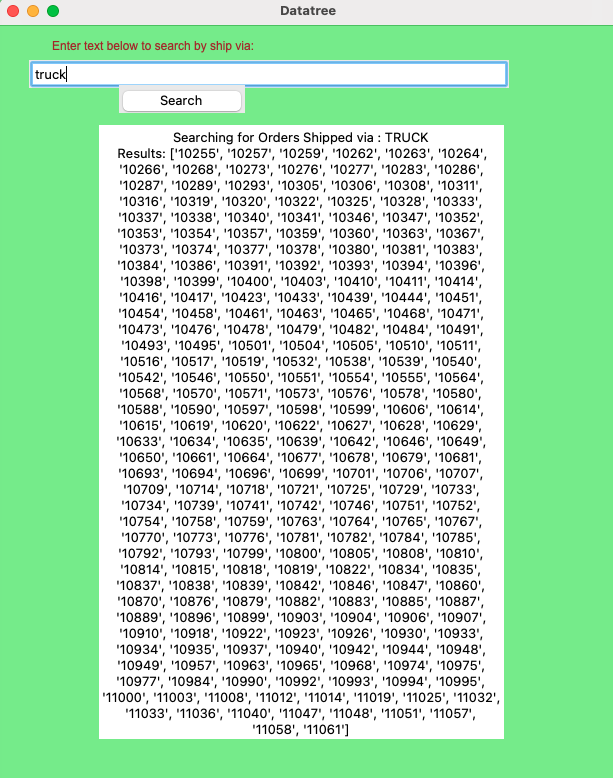
Search for orders with same required date:



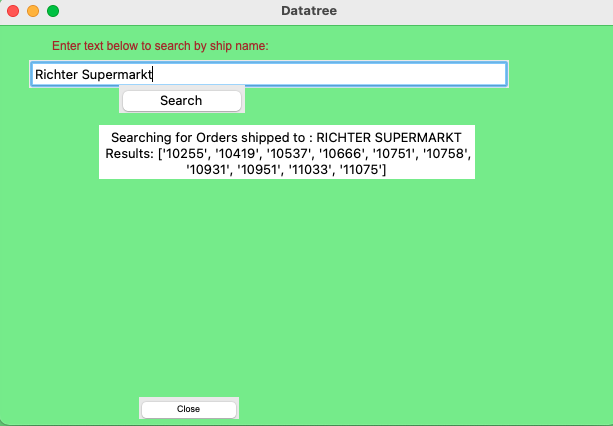
Searching for orders shipped on date



Searching for all orders shipped via truck



Searching for orders from same company:



Works Cited

Amazon Web Services. (n.d.). AWS Lambda Pricing. Retrieved March 7, 2023, from https://aws.amazon.com/lambda/pricing/

*Python enhancement proposals*. PEP 8 – Style Guide for Python Code. (n.d.). Retrieved March 5, 2023, from https://peps.python.org/pep-0008/

Project links:

Github: <https://github.com/CSC4110-Group1>