**Repo:**

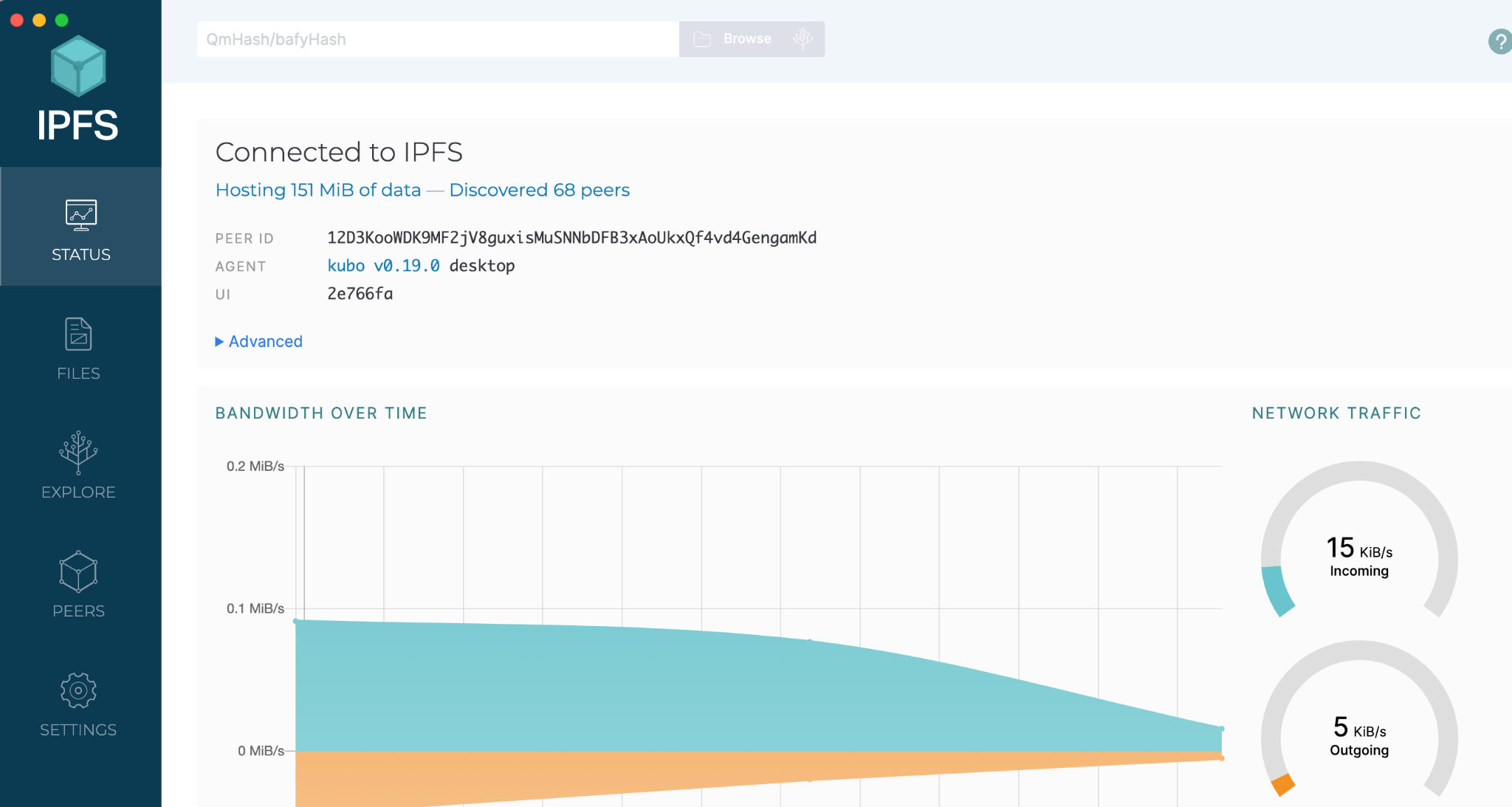
IPFSFinder is a centralized search engine for the IPFS network. Users can use IPFSFinder to search through IPFS files by content instead of by content identifiers (CID), similar to searching in other search engines like Google or Bing. IPFS members can make their files publicly accessible to users of IPFSFinder by downloading our agent software and allowing us to index them. Users can simply visit our website to begin searching and don’t need to download an IPFS node! This centralized approach to building a search engine on top of IPFS allows IPFSFinder to scale to the speed requirements of state-of-the-art search engines. We hope you enjoy it!

Below you will find detailed instructions on installing and running all of the components of IPFSFinder. The main components are

* Agent software: used by IPFS members to make their files public and searchable by users.
* Inverted Index Service: server responsible for receiving inverted indexes from the agent software and merging them to the global index stored in our database.
* Database: where the global index is stored. Client queries are solved using the information stored here.
* Client software: web interface that allows clients to search through IPFS content by content.

**Detail instructions on how to execute (step by step):**

1. **Agent software**
2. Open the IPFS software on your local machine and import the file which you want to be indexed.



1. Setup backend of agent software, use command. The backend server is listening on localhost:3001

npm install

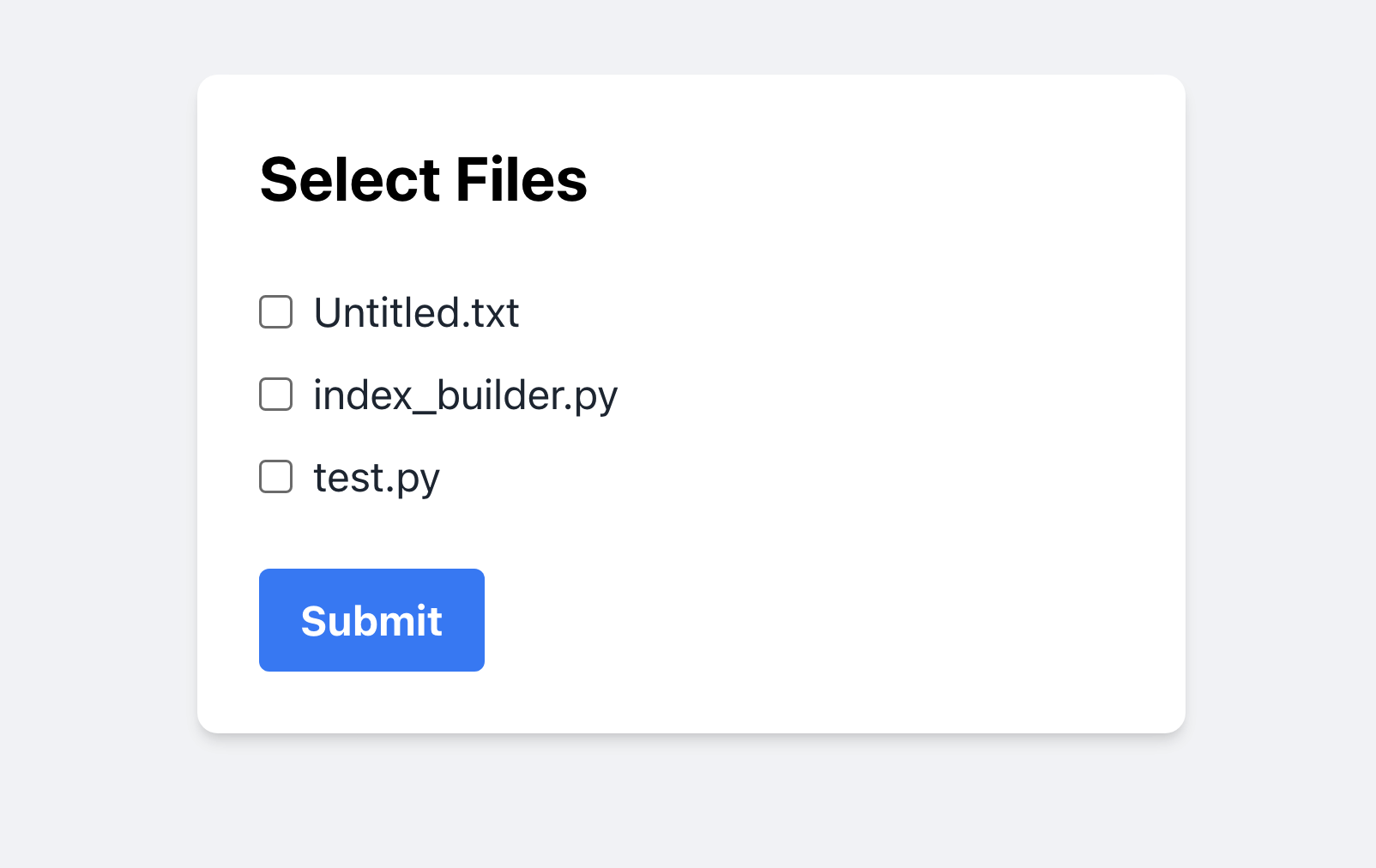
nodemon

1. Setup frontend of agent software, use command. The frontend server is listening on localhost:3000

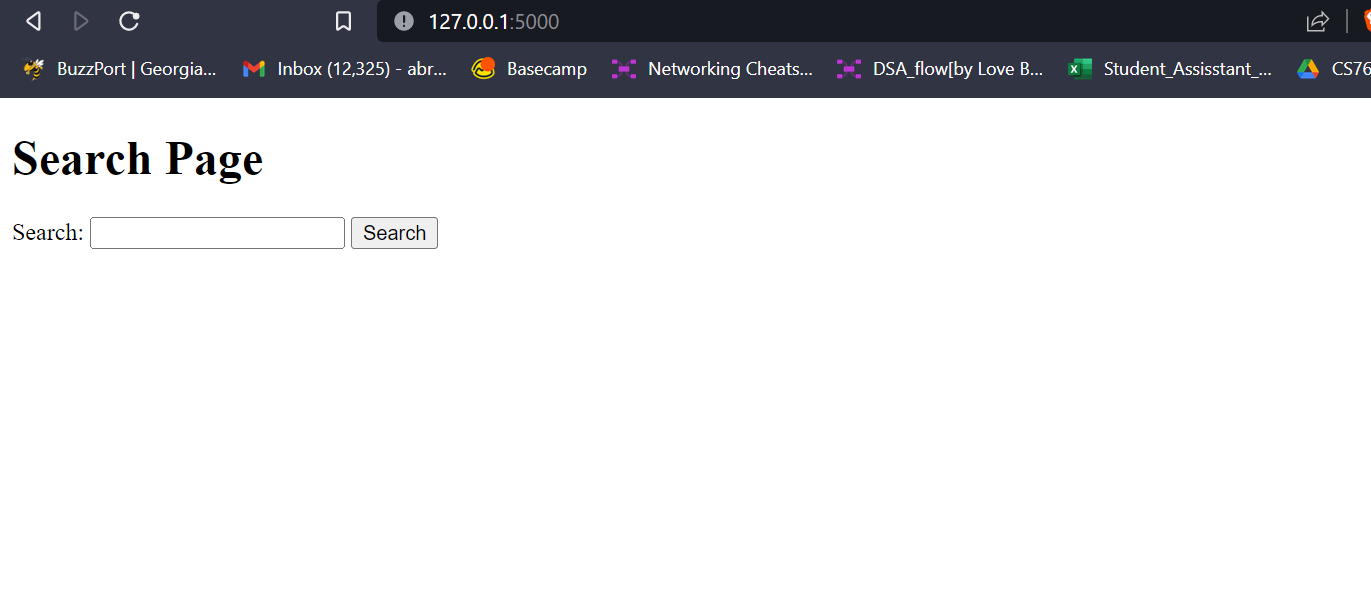
npm install

npm run start

1. Choose the file you want to be indexed.



1. **Inverted Index Service**
   1. Install python 2.8 or later
   2. Install mysql.connector requirement by running:
      1. pip install mysql-connector-python
   3. Navigate to the db\_updater.py location and run:
      1. python db\_updater.py -ip localhost -port\_num 5100
      2. make sure to use localhost and port number 5100 when running the server. The agent software is hard-coded to route messages to this address.
      3. After running this file, the agent software should be able to send inverted indexes that end up stored in the database
   4. Use the logs.log file to see the server connections in progress
   5. Run the testbd.py to print the current contents in the database. Users can run the testbd.py file before and after using the agent software to ensure the server can receive the JSON object and push it into the database. To run the file, navigate to the testbd.py location and run:
      1. python testbd.py
2. **Client software-database**
3. The database details have to be edited with [your-database-details]
4. In this repo, we used MySQl db client provided by Microsoft Azure
5. The required changes are in search\_logic() function in search.py
6. Update the parameters of the conn = mysql.connector.connect() with your db username, password, host name, database name
7. Lastly, change the table name in the below query in line 39; Note: in our case “keywords” is the table name
8. sql="SELECT \* FROM keywords WHERE keyword LIKE '%{}%'".format(input)
9. **Client server**
   1. Initialize client side backend using command: *python3 search.py*
   2. Load client side webpage using the url returned from the command above to launch the frontend locally. URL: <http://127.0.0.1:5000/>
   3. Type keyword in search bar



Example result:

