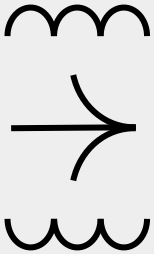
Google Developer Group
Sheridan College

Welcome Everyone

We're excited to have you here!

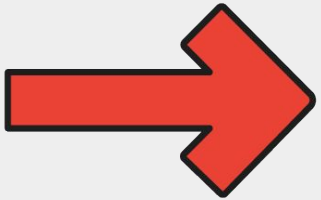
Please be seated. We will begin at 5:30 PM



Google Developer Group
Sheridan College

How to Build a Full Stack App: GitHub + Flask + Next.js + BigQuery

Workshop A - George Farag



Prerequisites

Python

python3 --version

https://www.python.org/downloads/

Node.js

node --version

npm --version

https://nodejs.org/en/download

git

git --version

https://git-scm.com/install/

Tech Stack

GitHub

- Version Control

Flask

- Backend API

Next.js

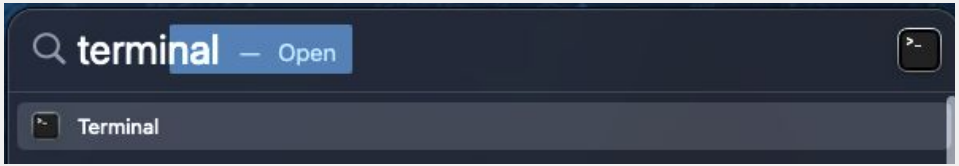
- Frontend

Google Cloud BigQuery

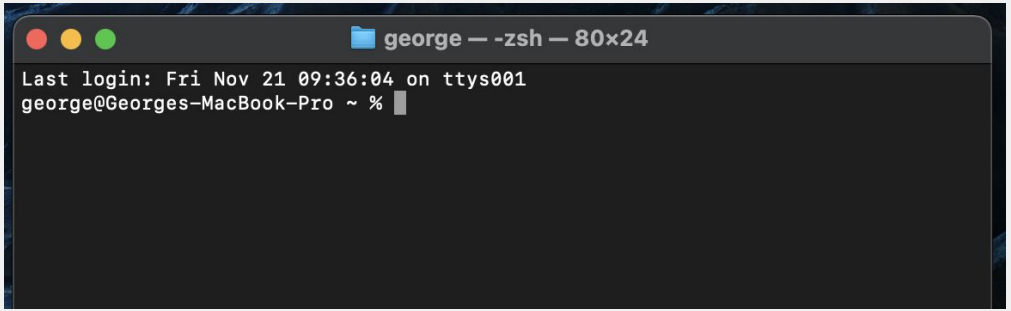
- Data Access

Step 1: Open Terminal

Open Terminal / Command Prompt on your laptop



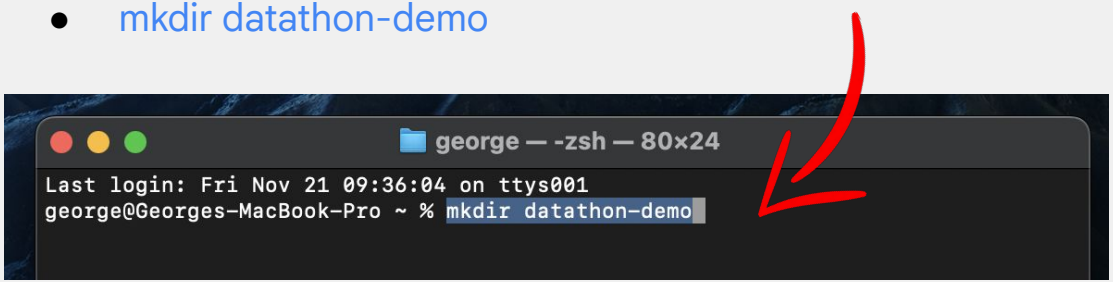
You should have this screen opened



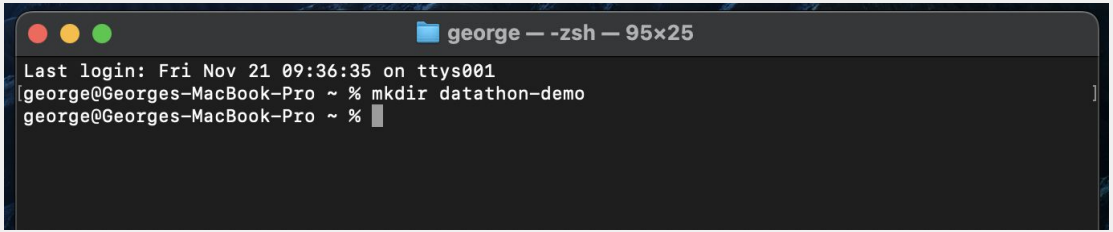
Step 2: Create Directory

Type the following into your Terminal to create a new folder

- mkdir datathon-demo



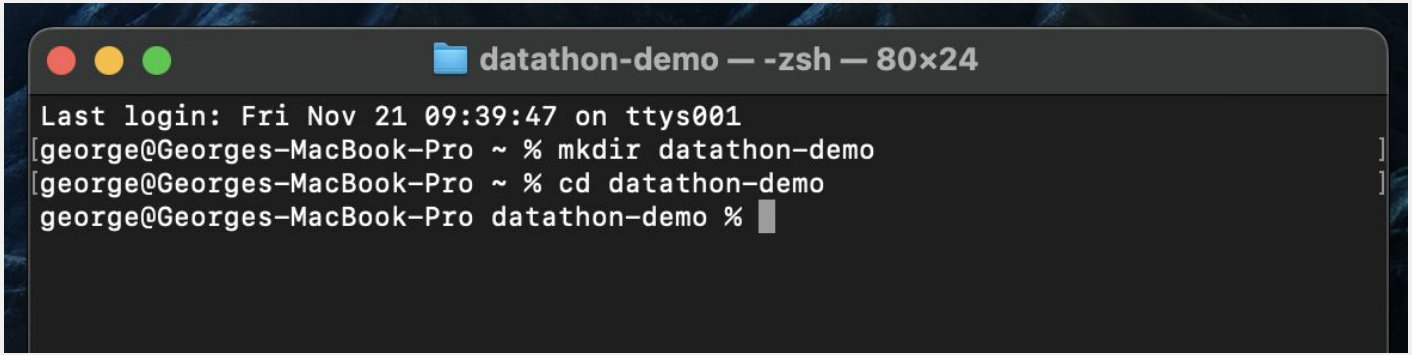
Press enter



Step 3: Navigate into Folder

Now use ‘change directory’ (cd) to navigate into your new folder

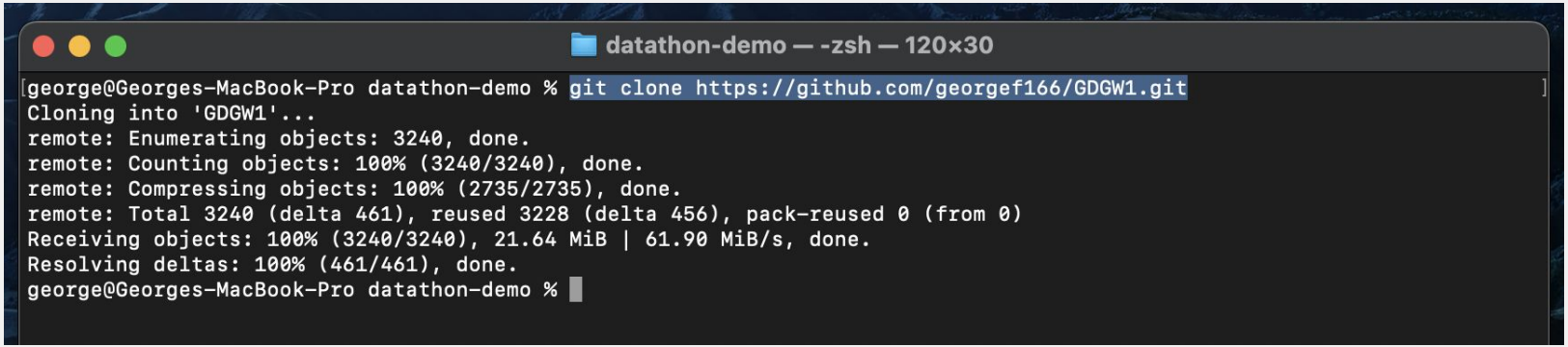
- cd datathon-demo



Step 4: Clone Repository

Use the following command to create a local copy of the repository

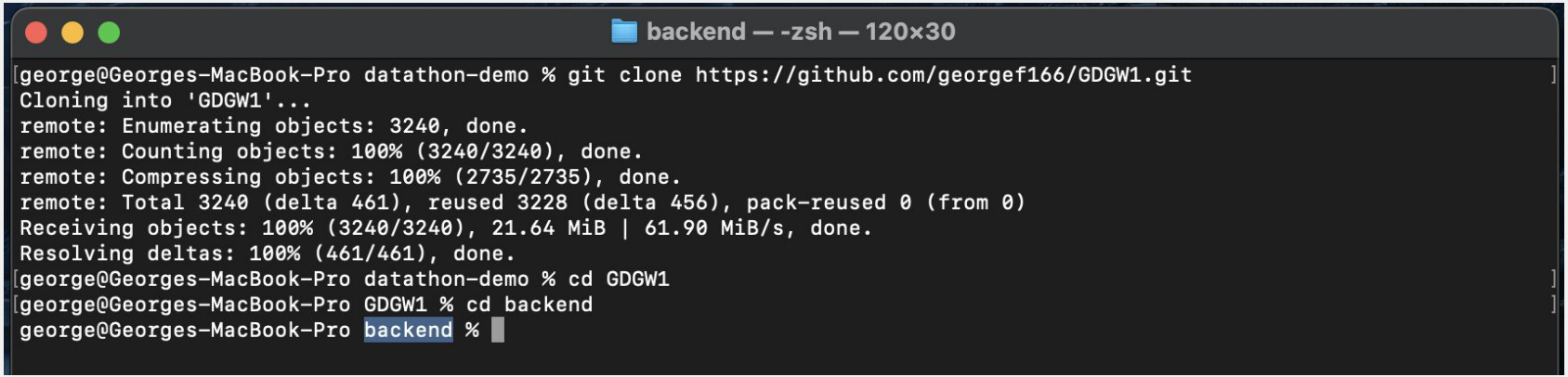
- git clone https://github.com/georgef166/GDGW1.git



Step 5: Navigate to Backend

Use ‘change directory’ (cd) twice to navigate into the backend folder

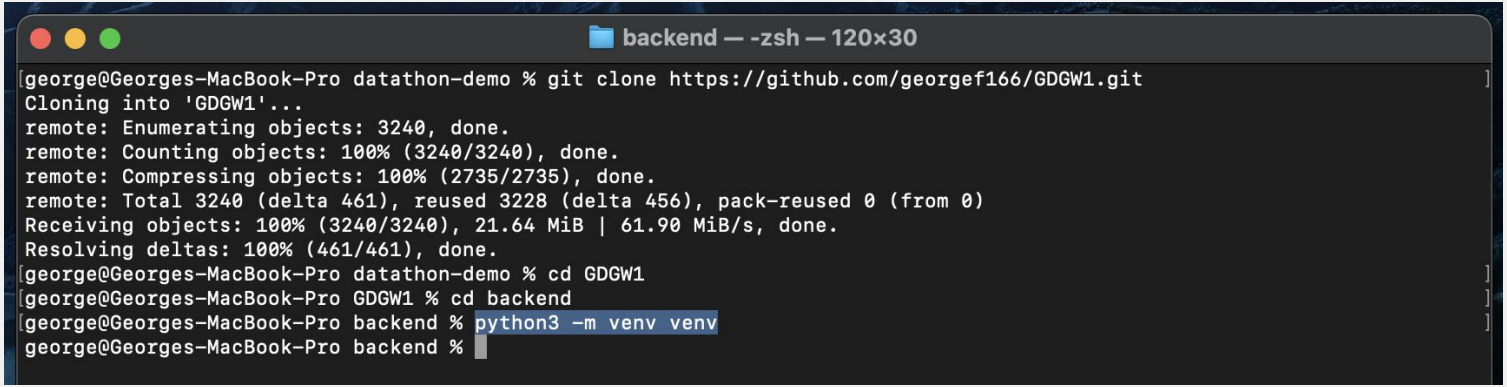
- cd GDGW1
- cd backend



Step 6: Initialize VENV

Initialize your virtual environment

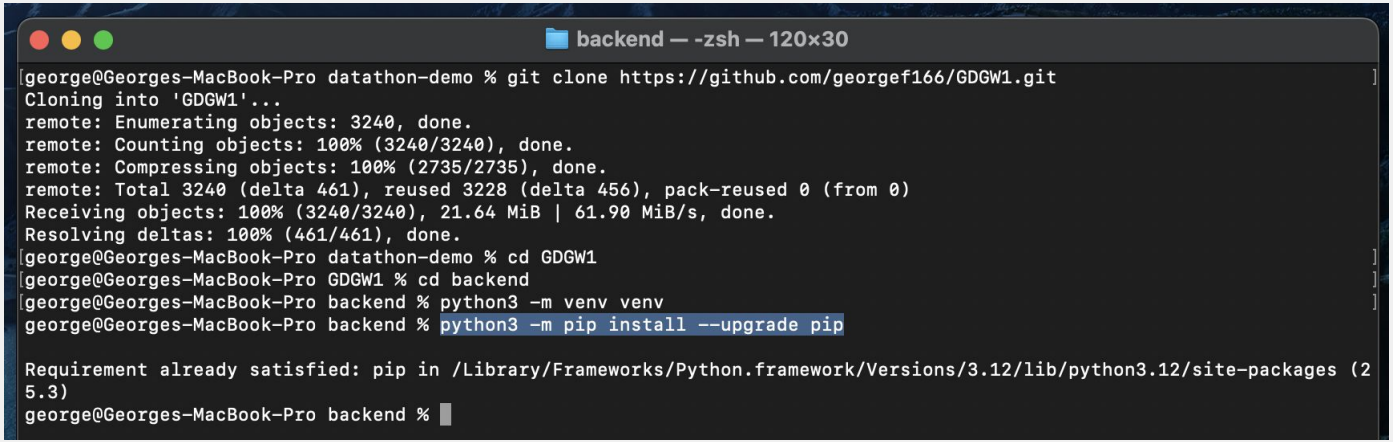
- python3 -m venv venv



Step 7: Install/Update pip

Ensure you have the standard package manager for Python

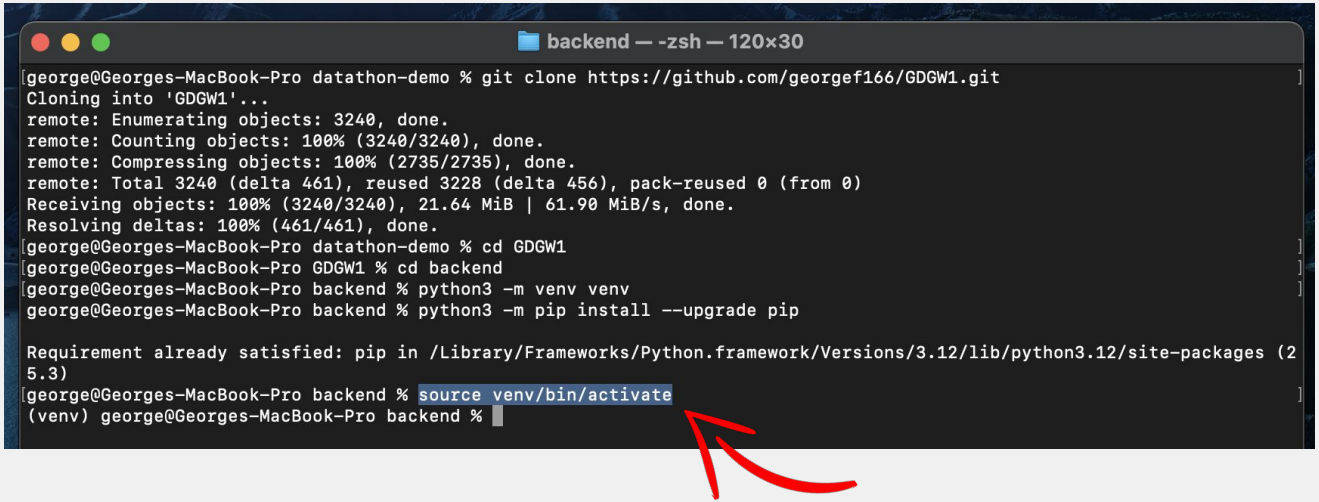
- python3 -m pip install --upgrade pip



Step 7: Activate Virtual Env

Type the command based on your platform to activate the venv

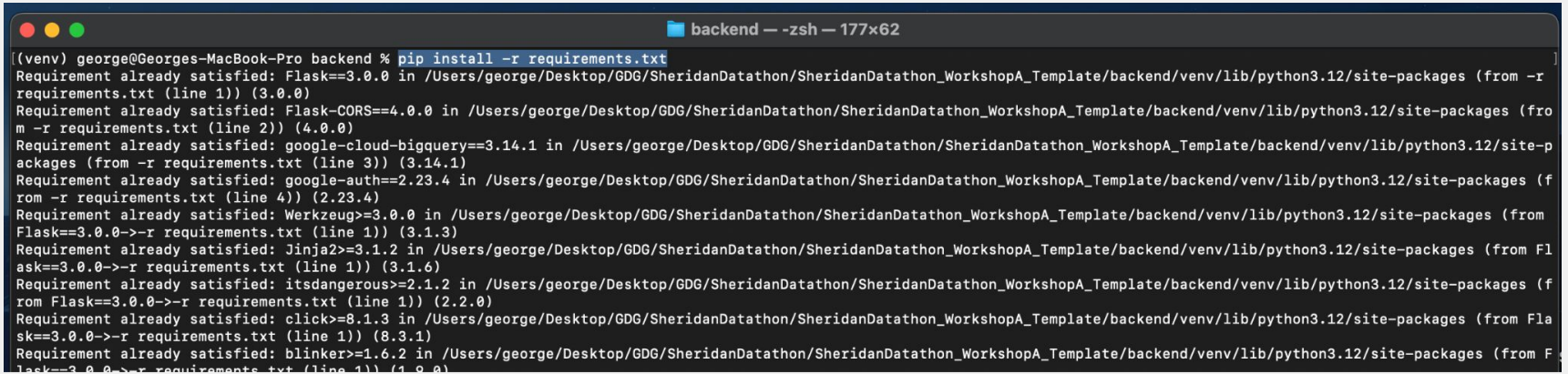
- `source venv/bin/activate` (Mac/Linux)
- `.\venv\Scripts\activate.bat` (Windows)



Step 8: Install Dependencies

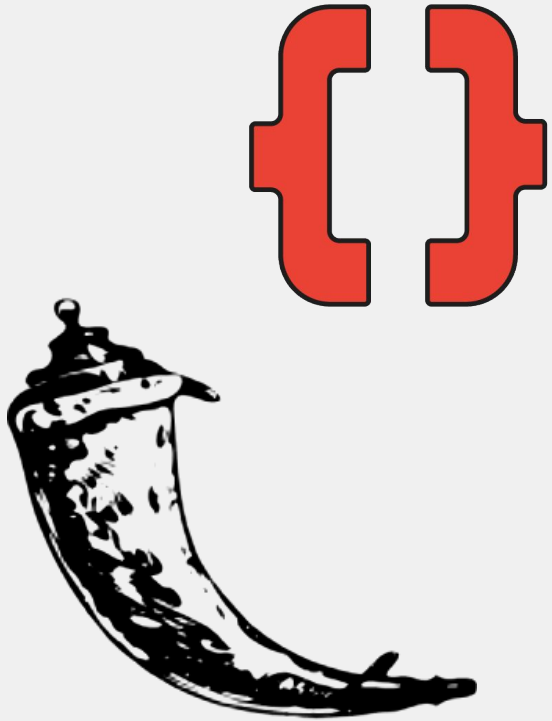
Type the command based on your platform to activate the venv

- `pip install -r requirements.txt`



Open VS Code - app.py

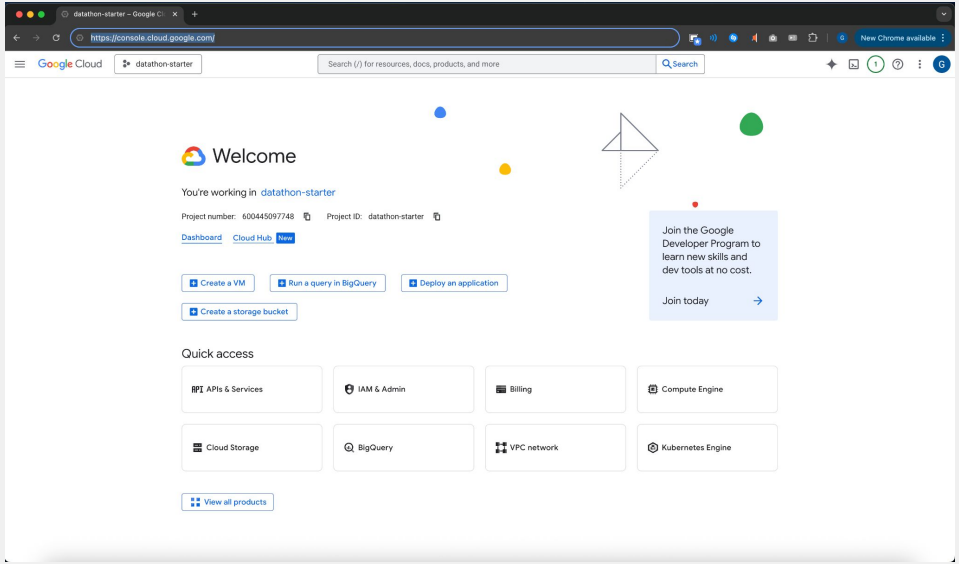
```
from flask import Flask, jsonify
from flask_cors import CORS
from google.cloud import bigquery

app = Flask(__name__)
CORS(app)

@app.route("/api/bq")
def bigquery_test():
        client = bigquery.Client()
        query = """
        SELECT name, gender
        FROM `bigquery-public-data.usa_names.usa_1910_current`
        LIMIT 5"""
        rows = client.query(query).result()
        return {"data": [dict(r) for r in rows]}

if __name__ == "__main__":
        app.run(port=5000, debug=True)
```



Step 9: Google Cloud Console

Navigate to the following link in your web browser (preferably Chrome)

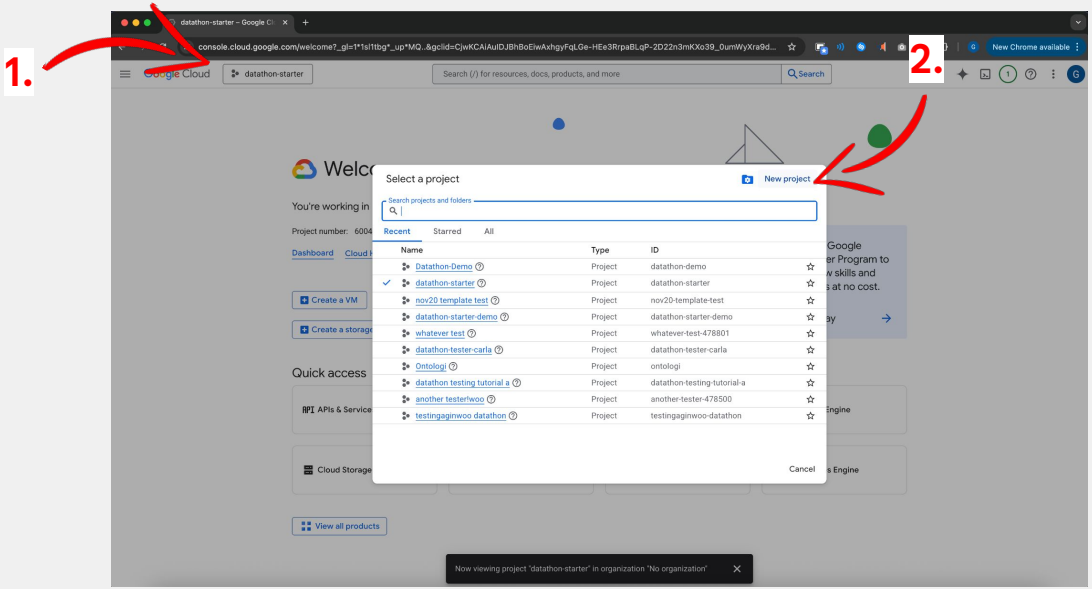
- console.cloud.google.com



Step 10: New Project

Press the New Project button in the top navbar

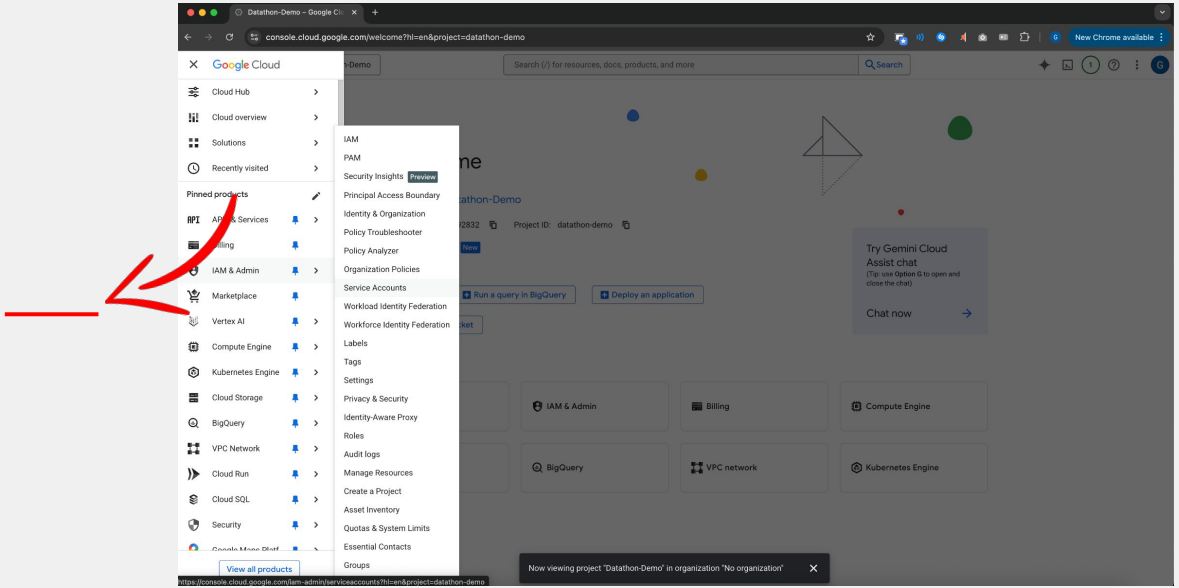
- Ensure the new project is selected upon creation, as by default it will not



Step 11: Service Account

Navigate to Service Accounts

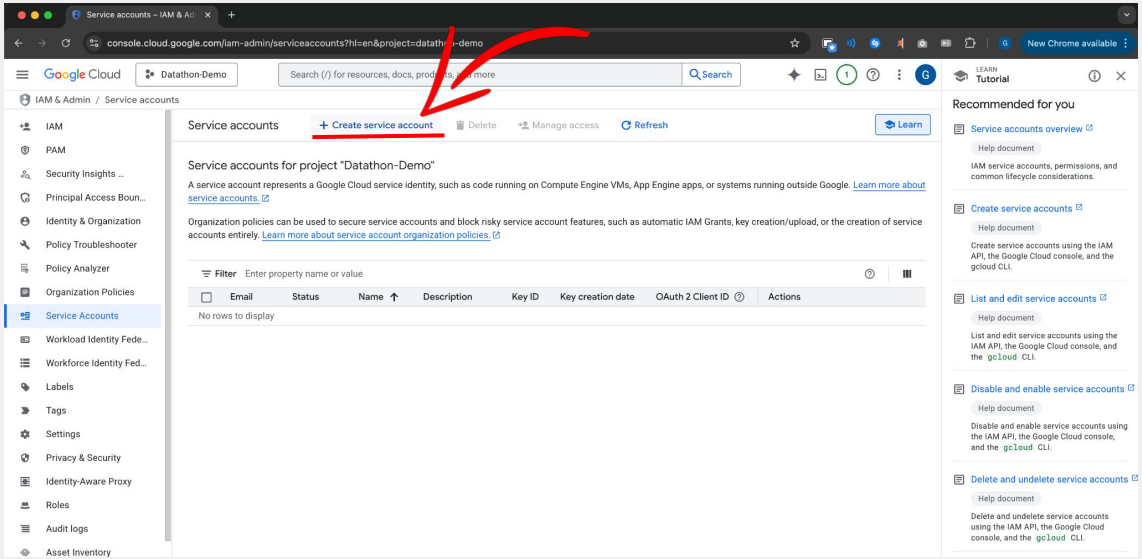
- IAM & Admin -> Service Accounts



Step 12: Create Account

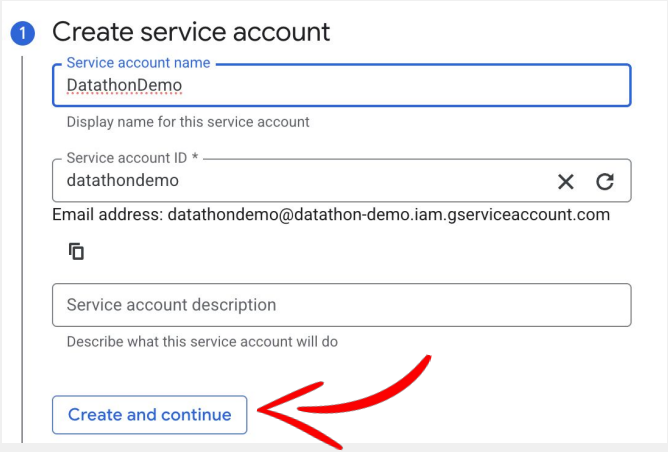
Create a Service Account

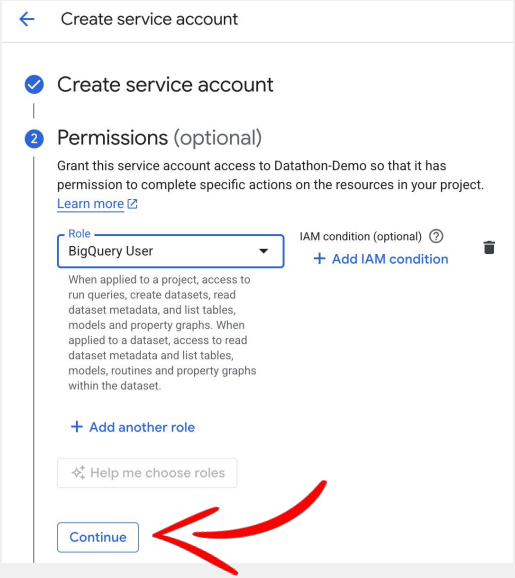
- Service Accounts -> Create Service Account

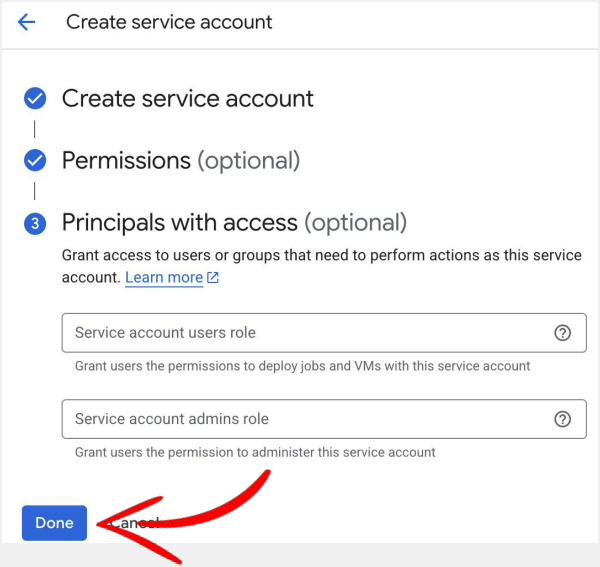


Step 13: Account Permissions

1. **Create service account ->** Enter any name -> **Create and continue**
2. **Permissions ->** BigQuery User -> **Continue**
3. **Principals with access ->** *leave empty* -> **Done**



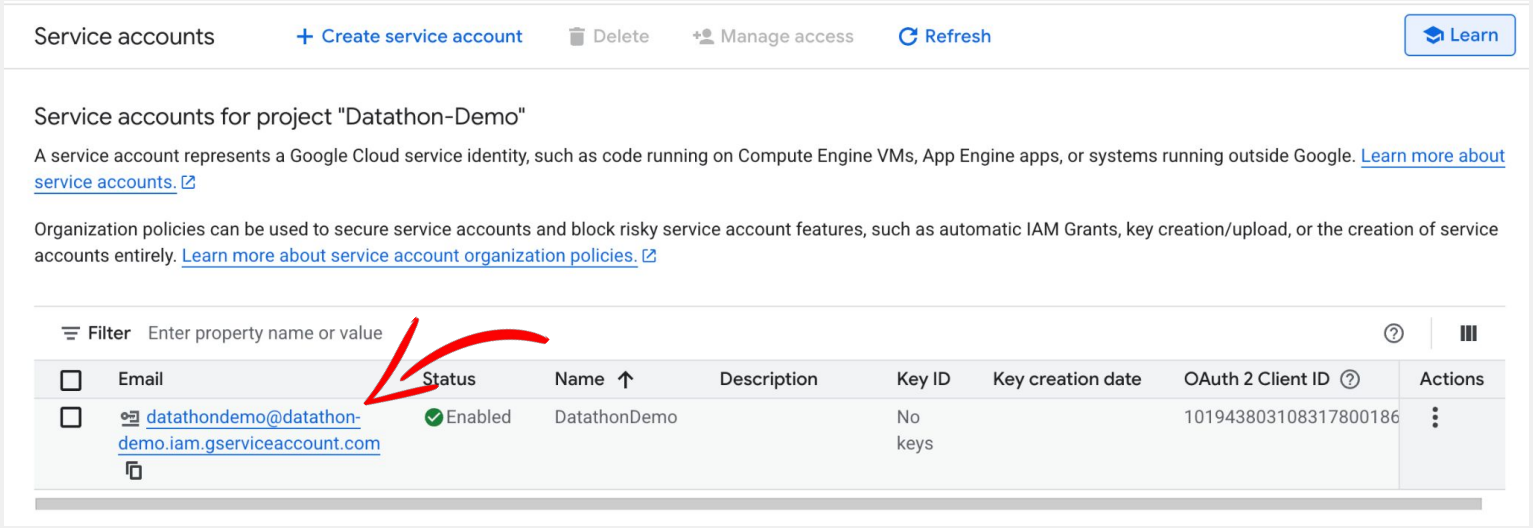




Step 14: Generate a Key

Select Service Account

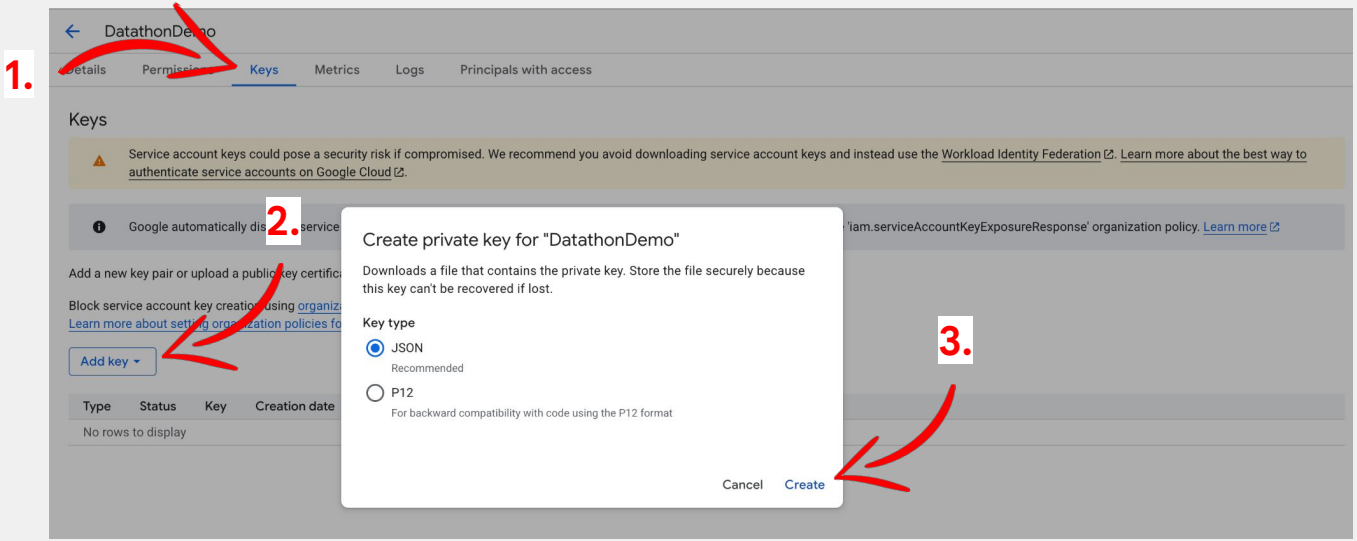
- Click on newly created Service Account



Step 15: Generate a Key

Add a JSON Key

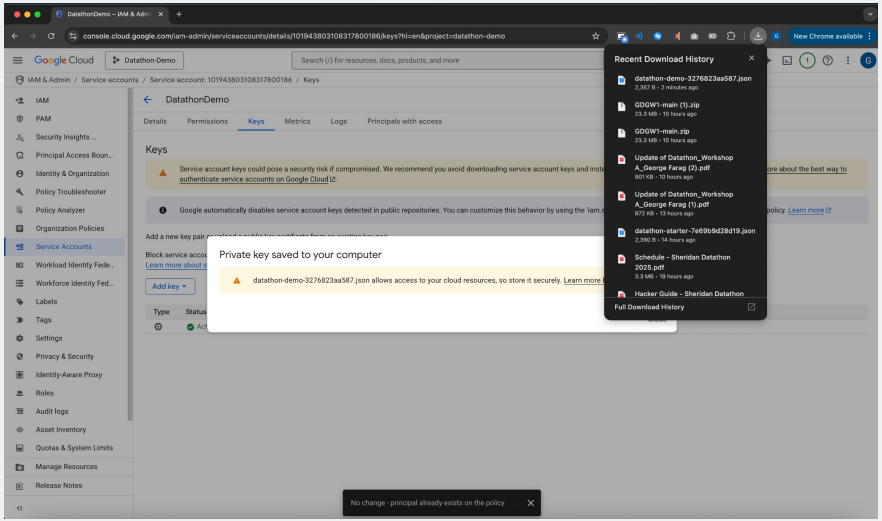
- Keys -> Add Key -> Create New -> JSON -> Create

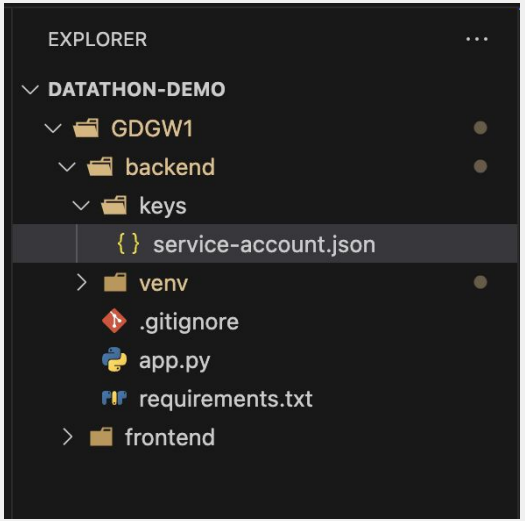


Step 16: service-account.json

Add a JSON Key

- Ensure JSON key was downloaded, and open the .json
- Copy and paste downloaded content in /backend/keys/service-account.json

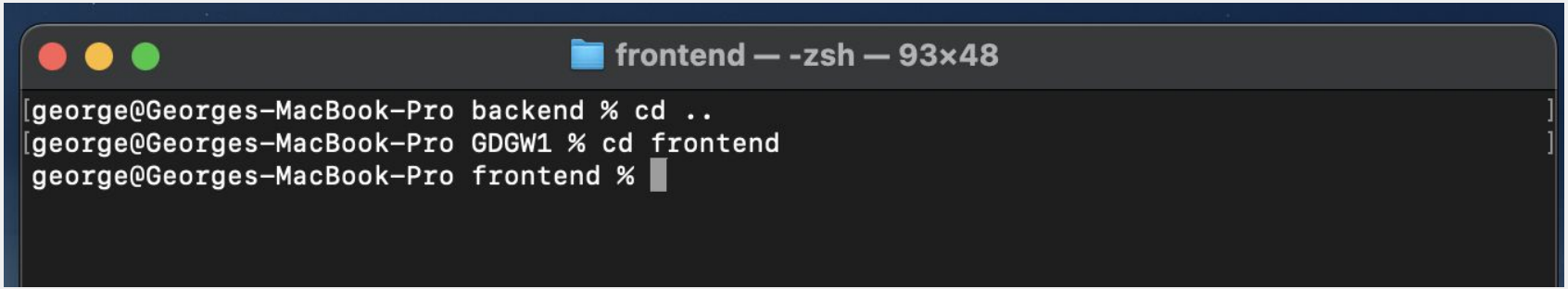




Step 17: Navigate to Frontend

Use ‘change directory’ (cd) to navigate into the frontend folder

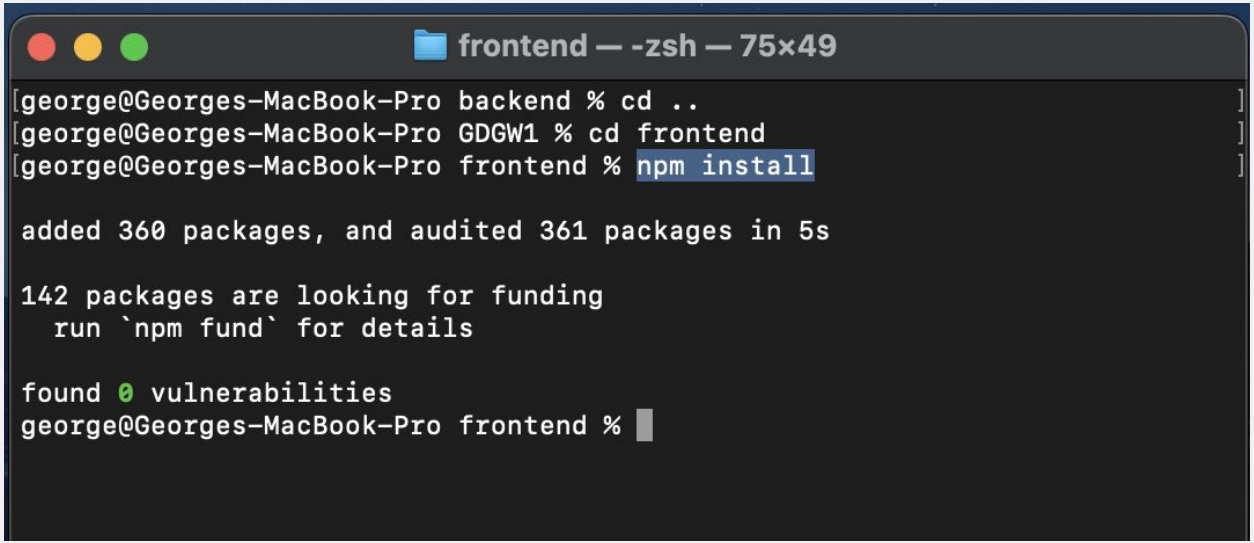
- cd ..
- cd frontend



Step 18: Node Packages

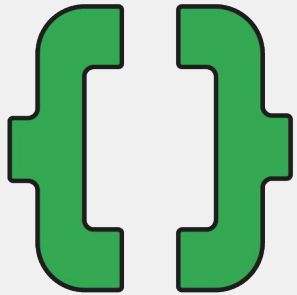
Install Required Node Packages with Node Package Manager

- npm install



Open VS Code - page.tsx

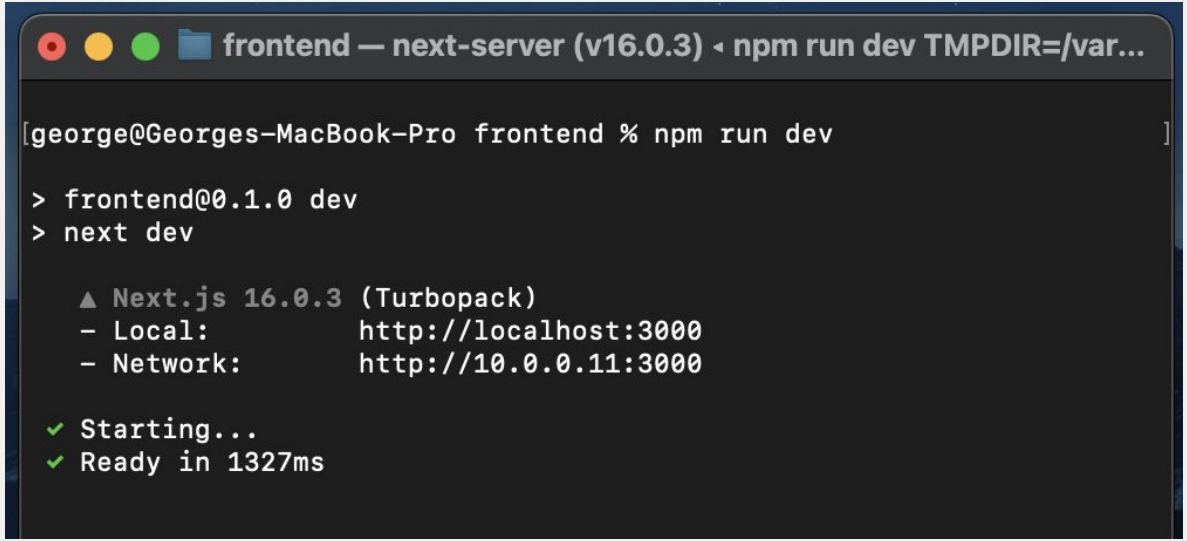
```python
from flask import Flask, jsonify
from flask_cors import CORS
from google.cloud import bigquery

app = Flask(__name__)
CORS(app)

@app.route("/api/bq")
def bigquery_test():
        client = bigquery.Client()
        query = """
        SELECT name, gender
        FROM `bigquery-public-data.usa_names.usa_1910_current`
        LIMIT 5"""
        rows = client.query(query).result()
        return {"data": [dict(r) for r in rows]}

if __name__ == "__main__":
        app.run(port=5000, debug=True)
```



Step 19: Run the Frontend

Run the Frontend

- npm run dev



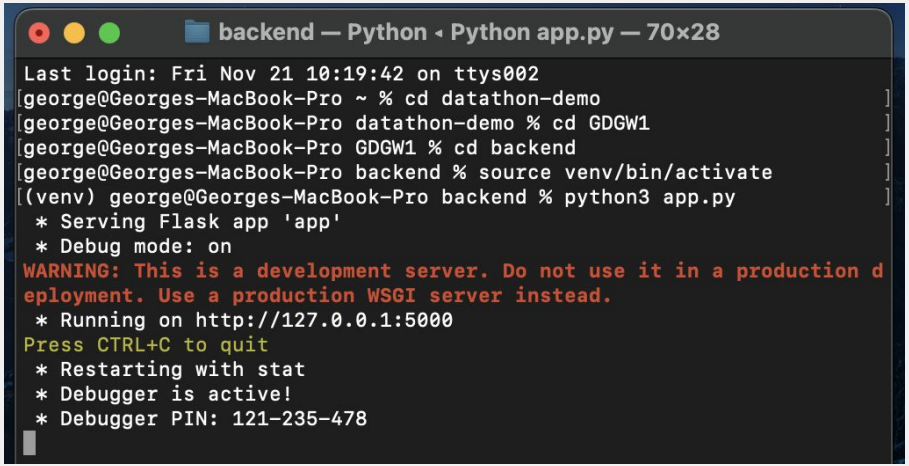
Step 20: Run the Backend

Open New Terminal window → Navigate to Backend
Activate venv → Run the Backend
Type the command based on your platform to activate the venv

- `source venv/bin/activate` (Mac/Linux)
- `.\venv\Scripts\activate.bat` (Windows)

Run the Backend

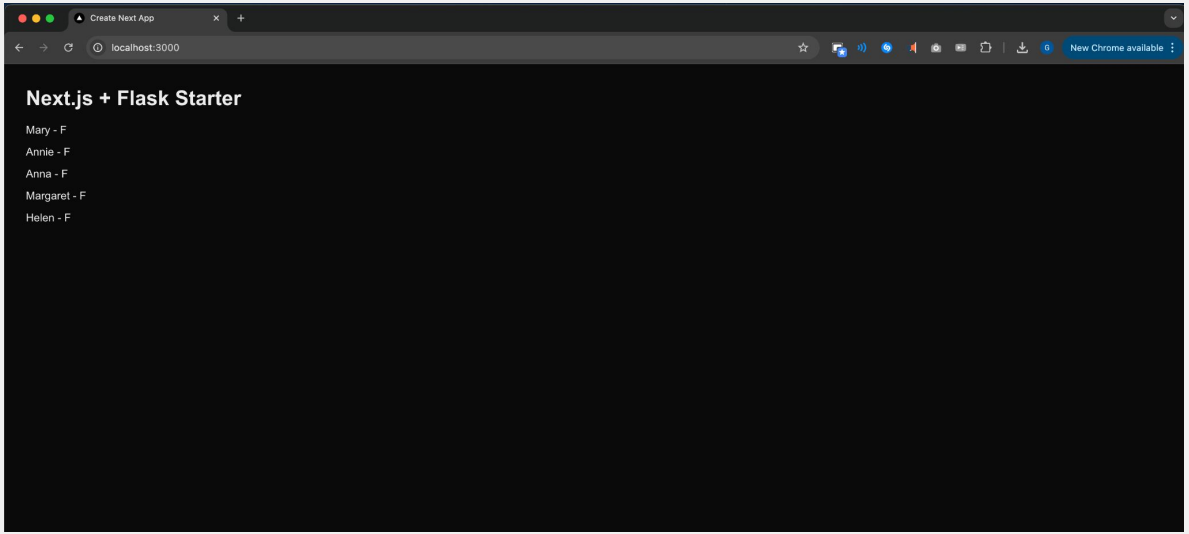
- python3 app.py



Step 21: Open your WebApp

localhost:3000 - Default location where Next.js hosts the Frontend

- http://localhost:3000

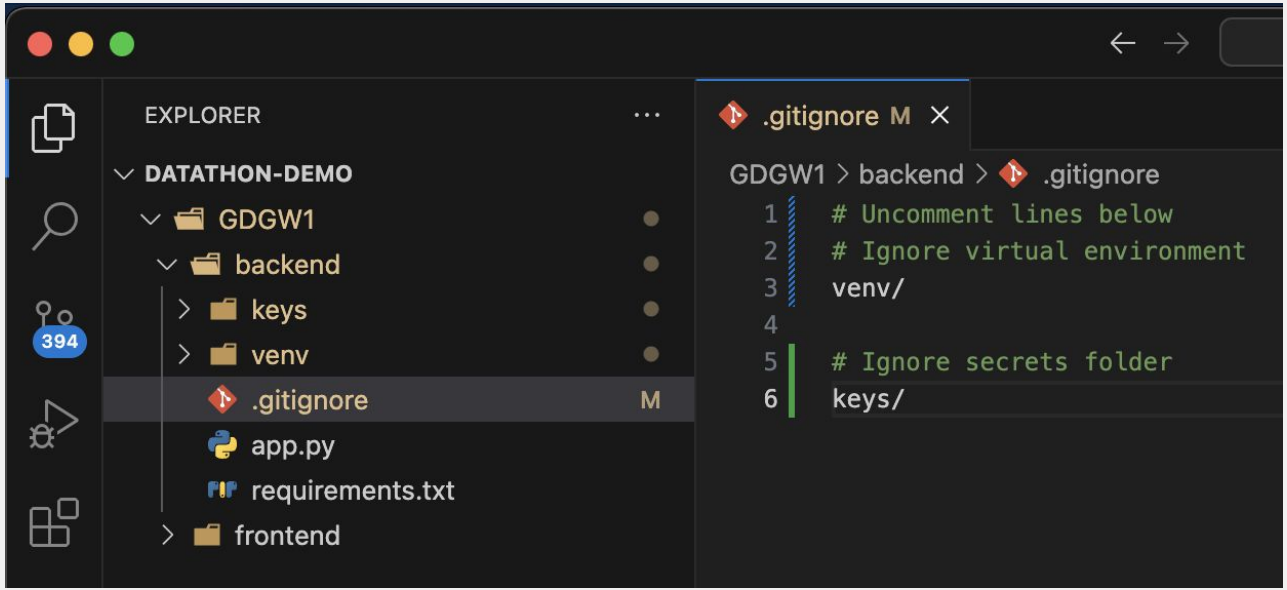


Step 22: .gitignore

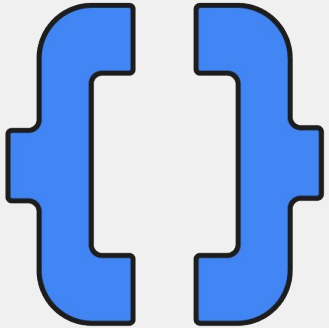
Locate the .gitignore in /backend

Ensures you do not push anything you do not want public

- Select All
- cmd + /

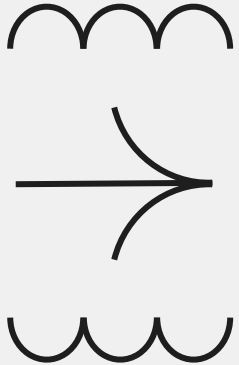


Recap



Flow

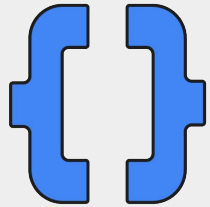
- User interacts with Next.js
- Next.js sends requests to Flask
- Flask authenticates using service account
- Flask sends SQL query to BigQuery
- BigQuery returns results
- Flask responds to frontend
- Frontend displays result



Execution

- Start Backend - Flask -> venv -> python3 app.py
- Start FrontEnd - Next.js -> npm run dev

Google Developer Group
Sheridan College



Thank You!

We hope you enjoyed the presentation!