**IT Capstone Documentation**

# Contents

Contents 1

Overview 1

Setting Up Jupyter Notebook 1

Setting Up Jupyter Scheduler 3

Setting Up Atlas MongoDB 4

Connecting Python Code to MongoDB 4

Python Script Description 5

Streamlit Dashboard 5

# Overview

This guide walks you through the process of setting up the Jupyter Notebook, a powerful open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It also covers the configuration of the Jupyter Scheduler, a tool for scheduling and automating your Jupyter notebooks.It provides step-by-step instructions on how to establish a connection between Python code and Atlas MongoDB, a fully managed cloud database. It also include Streamlit setup, an open-source app framework for Machine Learning and Data Science teams. Streamlit lets you create interactive Dashboards/web applications for data exploration and visualization. It’s a fast, user-friendly way to create and share data-driven web apps.

This guide is a great resource for those looking to integrate these technologies and enhance their data analysis and visualization capabilities."

# Setting Up Jupyter Notebook

You can install Jupyter Notebook on your computer using pip, which is a package manager for Python. You will need to open your computer’s command prompt (Windows) or terminal (macOS/Linux).

Then, you can install Jupyter Notebook by typing the following command and pressing enter:

1. **Install Jupyter Notebook**: You can install Jupyter using pip:

|  |
| --- |
| # This command installs the Jupyter Notebook app via pip, Python's package installer  pip install notebook |

Once the installation is complete, you can launch Jupyter Notebook by typing the following command and pressing enter:

1. **Launch Jupyter Notebook**: You can start the notebook server from the command line using the following command:

|  |
| --- |
| # This command launches the Jupyter Notebook application  jupyter notebook |

This will open a new window in your default web browser that shows the Notebook Dashboard, where you’ll be able to create and manage your notebooks.

1. **Create a New Notebook**: From the dashboard, you can create a new notebook by clicking on the “New” button and selecting “Python 3” or your preferred kernel.

# Setting Up Jupyter Scheduler

Setting up the Jupyter Scheduler with Jupyter Lab involves a few steps. Here’s a detailed guide:

1. **Install Jupyter Scheduler**: You can install the Jupyter Scheduler library using pip, which is a package manager for Python. Open your command prompt or terminal and type the following command:

|  |
| --- |
| # This command installs the Jupyter Scheduler library via pip  pip install jupyter\_scheduler |

1. **Restart JupyterLab**: After installing the Jupyter Scheduler, you may need to restart JupyterLab for the changes to take effect.
2. **Create a Notebook Job:** Once JupyterLab is restarted, you can create a notebook job. Right-click on your Jupyter Notebook in the file browser and locate the “Create Notebook Job” option.
3. **Configure the Job:** Provide a job name and select the output formats. If your notebook requires any parameters, provide them as well.
4. **Schedule the Job:** Select the “Run on schedule” option. You can choose to run your notebook at specified intervals such as daily, weekly, or monthly.
5. **Test the Schedule:** Run an instance of the schedule on demand to see if it executes and creates the desired outputs.

# Setting Up Atlas MongoDB

1. **Create an Account:** Go to the MongoDB website and create an account.
2. **Create a New Project:** Click on “New Project”, name it “Finance\_Data”, and create it.
3. **Build a New Cluster:** Click on “Build a Cluster”, select the free plan, and name the cluster “Cluster0”.
4. **Create a New Database:** Click on “Collections” -> “Add My Own Data” enter “Data\_Retrieve” as the database name and “Capstone” as the collection name.
5. **Add Users:** Go to “Database Access” under the “Security” tab in the left-hand menu. Add a new database user with a username and password.
6. **Give Team Access:** Go to “Organization Access” under the “Access Manager” tab. Add your team members and assign them roles.
7. **Connect to Your Cluster**: Click on “Connect” on your cluster. Choose “Connect your application”. Select Python as your driver and copy the connection string.

# Connecting Python Code to MongoDB

You can use the pymongo library to connect your Python code to MongoDB. Here’s a sample code snippet:

|  |
| --- |
| from pymongo import MongoClient  # Connect to MongoDB  client = MongoClient('mongodb+srv://username:password@cluster0.mongodb.net/Data\_Retrieve?retryWrites=true&w=majority')  db = client['Data\_Retrieve']  collection = db['Capstone'] |

Replace 'username:password' with your actual username and password.

# Python Script Description

The code written in **Save headlines to MongoDB.ipynb document** is using Python script that scrapes finance news using BeautifulSoup for a list of stock tickers from Yahoo Finance’s RSS feed. It then stores this data in a MongoDB collection. The get\_finance\_news function fetches the news data and returns it as a DataFrame. The main function calls get\_finance\_news for a list of tickers, converts the DataFrame to a dictionary, and then inserts each record into the MongoDB collection. The script then prints out all the records in the collection. The MongoDB connection is established using the pymongo library. The connection string includes the username and password for authentication and specifies the Data\_Retrieve database and Capstone collection.

Replace 'username:password' with your actual username and password in the connection string. Also, ensure that the IP address of the machine running this script is whitelisted in your MongoDB Atlas network access settings.

# Streamlit Dashboard

Streamlit dashboard is a python library that helps to build an interactive and user-friendly dashboard for users to access information.

**1: Install** **Streamlit :** You can install the Streamlit library using pip, which is a package manager for Python. Open your command prompt or terminal and type the following command:

2: **Opening the Streamlit App**

* Open your command line interface (CLI), such as Command Prompt (Windows) or Terminal (Mac/Linux).
* Navigate to the directory where the Streamlit app script (**app.py**) is located. Use the cd command to change directories.
* Once in the project directory, run the following command to launch the Streamlit app:

* After running the command, you will see output similar to:

#### 3: **Accessing the Streamlit App:**

* Open a web browser (e.g., Chrome, Firefox).
* In the address bar, enter the Local URL provided by Streamlit:

#### 4 **: Interacting with the Streamlit App**

* Once the Streamlit app is opened in the browser, you can interact with the app's interface based on its functionality.
* Use the provided input fields, sliders, buttons, etc., to explore the app's features and visualize the data.

#### **Features Overview:**

Our Streamlit application offers the following features:

* Sentiment Analysis using Vader
* Article Summarization with the Newspaper library
* Stock Prices Retrieval using yfinance

### **Instructions:**

#### **Sentiment Analysis:**

1. Navigate to the " Sentimental Score of News " section of the application.
2. Enter the Start date and End date you would like to analyze in the provided text input box.
3. The dashboard will display the sentiment analysis using the Vader library.
4. The sentiment analysis results will be displayed, showing the sentiment score (positive, negative, neutral) of the entered text.

#### **Article Summarization:**

1. Go to the " Summarize News Article " section.
2. Enter the stock symbol and the date of the articles you want to summarize in the input field.
3. The summarized version of all the articles will be displayed, providing a concise overview of the main points.

#### **Stock Prices Retrieval:**

1. Head to the " Stock Price Visualization " section.
2. Enter the stock ticker symbol (e.g., AAPL for Apple Inc.) in the input field.
3. The historical stock prices for the specified stock and timeframe will be displayed in a table format.