Configuration Manual

MSc Research Project
Msc in Cybersecurity

Athipan Thiyada Sridharan
Student ID: x22135405

School of Computing
National College of Ireland

Supervisor: Apurva Vangujar

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Athipan Thiyada Sridharan
Student ID:	x22135405
Programme:	MSc in Cybersecurity
Year:	2023-24
Module:	Project
Supervisor:	Apurva Vangujar
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	850
Page Count:	9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Athipan Thiyada Sridharan
x22135405

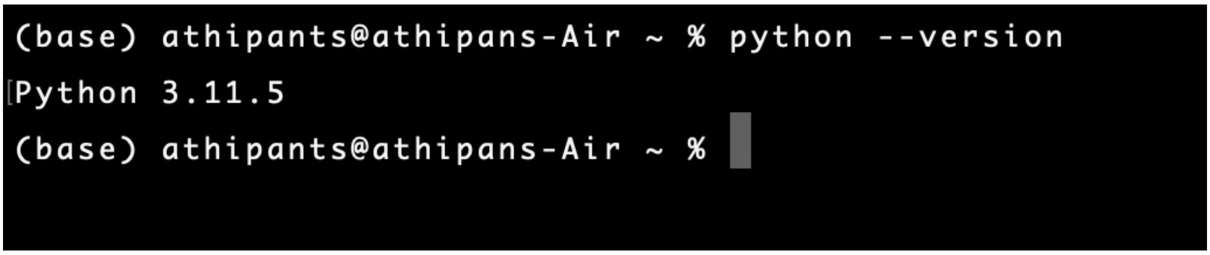
This guide provides step-by-step instructions for downloading, installing, and configuring the essential libraries and tools required for our project. The focus will be on Integrating the Random Forest model with the existing web application

1 Prerequisites

- **Operating System:** Any popular OS
- **Python:** Python 3.x
- **Node.js and npm:** Required for running and building the ReactJS frontend.
- **Machine Learning libraries:** Scikit-learn components
- **Flask and Libraries:** Flask and other Python libraries as listed in 'requirements.txt'.
- **Database:** Anyone database server like PostgreSQL, MySQL, or SQLite.

2 Python Environment Setup

The playground application uses Python as the backend, thus it is necessary to have Python 3 installed on the machine pyt (n.d.).



```
(base) athipants@athipans-Air ~ % python --version
[Python 3.11.5
(base) athipants@athipans-Air ~ %
```

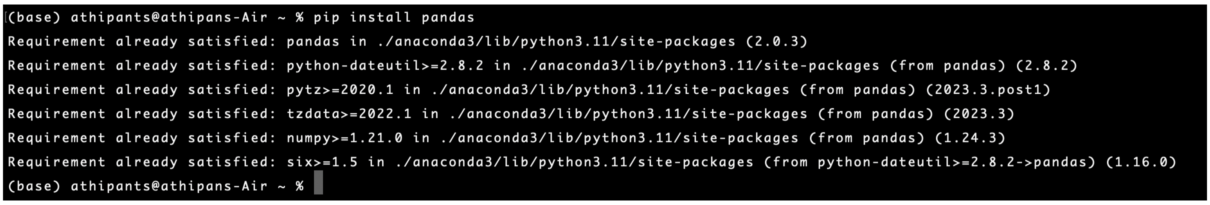
Figure 1: Python installation

3 Installation of Pandas

1. Open the terminal or command prompt.

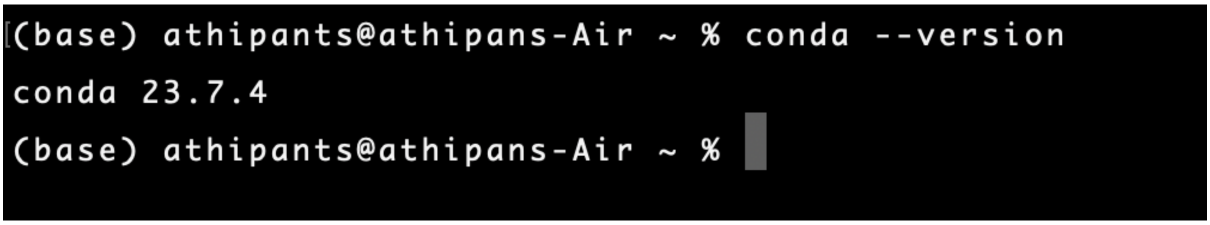
2. Install Pandas by executing ”pip install pandas”

3. Verify the installation by running ”conda –version” pan (n.d.)



```
(base) athipants@athipans-Air ~ % pip install pandas
Requirement already satisfied: pandas in ./anaconda3/lib/python3.11/site-packages (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in ./anaconda3/lib/python3.11/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in ./anaconda3/lib/python3.11/site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in ./anaconda3/lib/python3.11/site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in ./anaconda3/lib/python3.11/site-packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in ./anaconda3/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
(base) athipants@athipans-Air ~ %
```

Figure 2: Pandas Installation



```
(base) athipants@athipans-Air ~ % conda --version
conda 23.7.4
(base) athipants@athipans-Air ~ %
```

Figure 3: Pandas version check

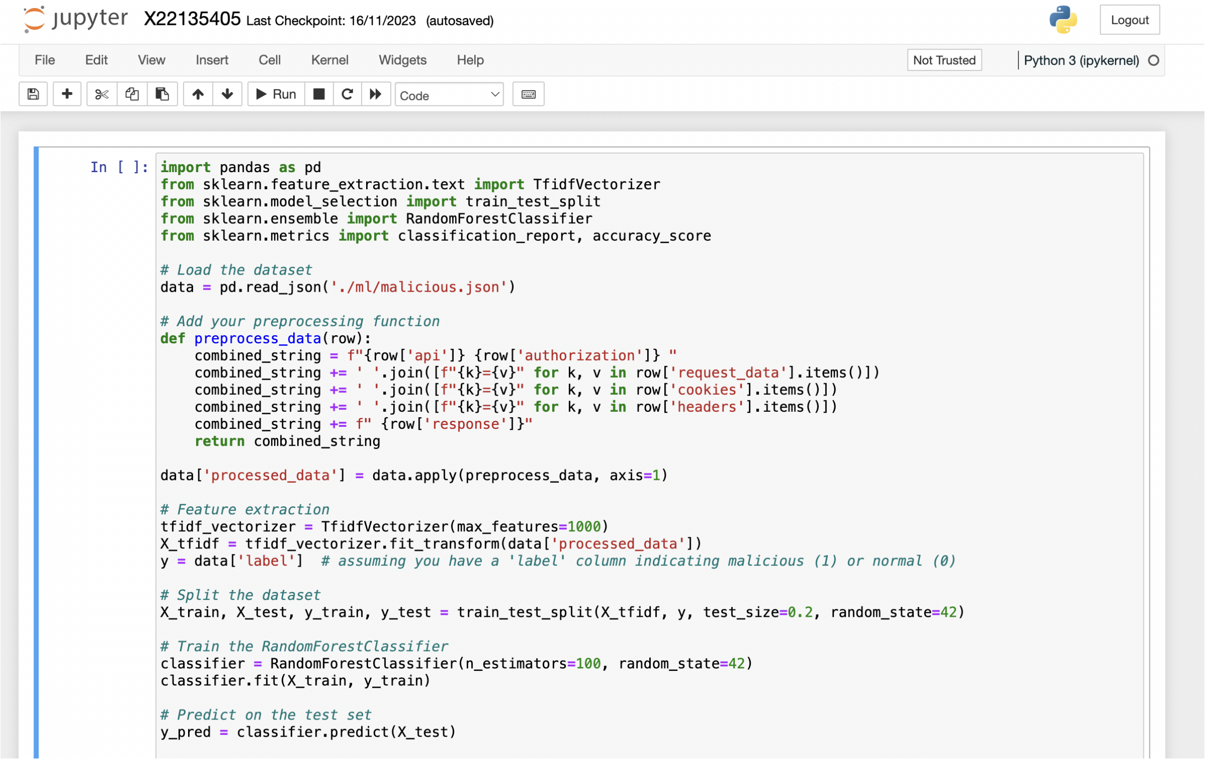
4. This installs the scikit-learn library which is the machine-learning library that is used for training our model.

5. Jupyter Notebook is also installed with this package, which is used to edit and modify the Python codes.

4 Installation of Required Python Packages for Flask Application

To set up the environment for your Flask application, you will need to install several Python packages. These packages provide the necessary functionality for your application's backend. Below is a step-by-step guide to install each of the required packages:

- **Flask:**
  - Flask is a micro web framework for Python, which is used to create the backbone of the web application fla (n.d.).
  - Installation: `pip install Flask`
- **Flask-CORS:**
  - Flask-CORS is an extension for handling Cross-Origin Resource Sharing (CORS), making cross-origin AJAX calls possible.



```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

# Load the dataset
data = pd.read_json('./ml/malicious.json')

# Add your preprocessing function
def preprocess_data(row):
    combined_string = f"{row['api']} {row['authorization']} "
    combined_string += ' '.join([f"{k}={v}" for k, v in row['request_data'].items()])
    combined_string += ' '.join([f"{k}={v}" for k, v in row['cookies'].items()])
    combined_string += ' '.join([f"{k}={v}" for k, v in row['headers'].items()])
    combined_string += f" {row['response']}"
    return combined_string

data['processed_data'] = data.apply(preprocess_data, axis=1)

# Feature extraction
tfidf_vectorizer = TfidfVectorizer(max_features=1000)
X_tfidf = tfidf_vectorizer.fit_transform(data['processed_data'])
y = data['label']  # assuming you have a 'label' column indicating malicious (1) or normal (0)

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2, random_state=42)

# Train the RandomForestClassifier
classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train, y_train)

# Predict on the test set
y_pred = classifier.predict(X_test)
```

Figure 4: Jupyter Notebook

  - Installation: `pip install Flask-CORS`

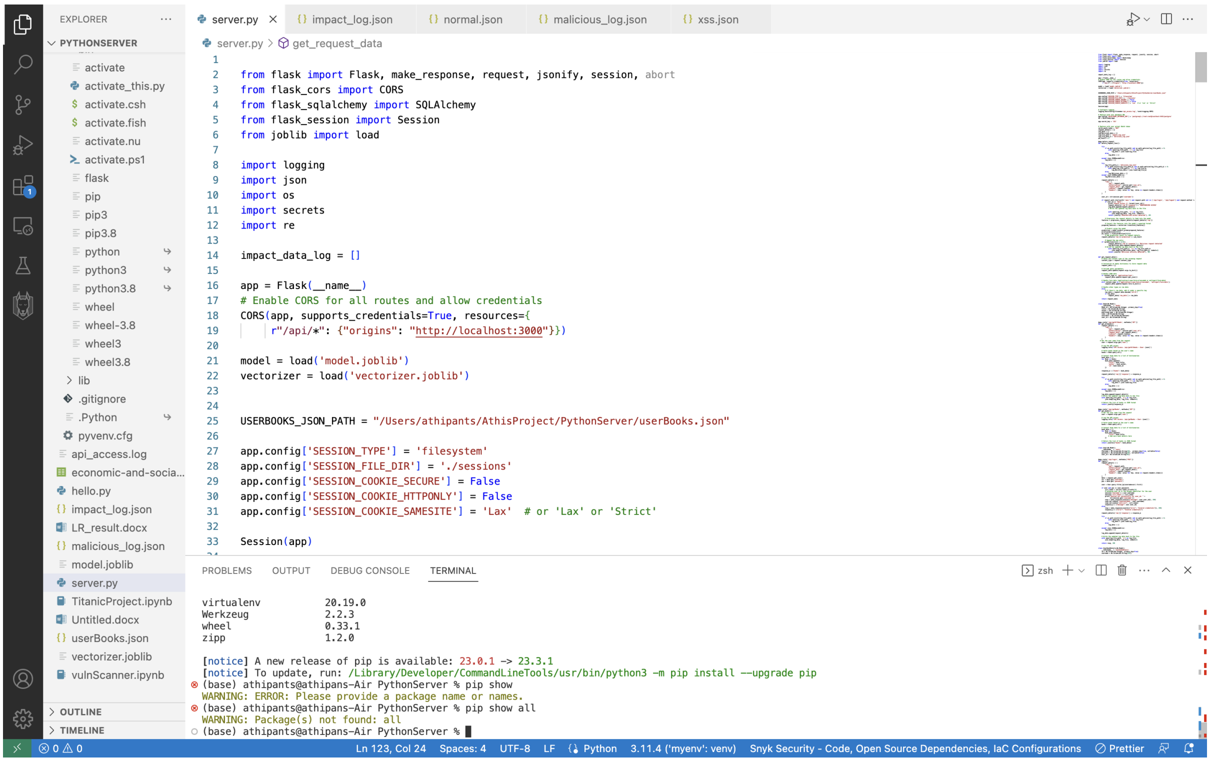
- **Flask-SQLAlchemy:**
  - Flask-SQLAlchemy is a SQL toolkit and Object-Relational Mapping (ORM) framework. This framework helps to connect with the DB and handle the requests to the DB .
  - Installation: `pip install Flask-SQLAlchemy`

- **Flask-Session:**
  - Flask-Session is an extension for Flask that adds support for server-side sessions to your application.
  - Installation: `pip install Flask-Session`

- **Joblib:**
  - Joblib is used for saving and loading Python objects that make use of NumPy data structures, efficiently.
  - Installation: `pip install joblib`
  - This library is used for integrating the trained model with out web application.

- **Logging:**
  - Logging module is part of the Python Standard Library, used for tracking events that happen when some software runs.

– Installation: Not required as it comes with Python.



```python

from flask import Flask, make_response, request, jsonify, session, abort
from flask_cors import CORS
from flask_sqlalchemy import SQLAlchemy
from flask_session import Session
from joblib import load

import logging
import json
import os
import secrets
import re

impact_data_log = []

app = Flask(__name__)
# Enable CORS for all routes and allow credentials
CORS(app, supports_credentials=True, resources={
        r"/api/*": {"origins": "http://localhost:3000"}})

model = load('model.joblib')
vectorizer = load('vectorizer.joblib')


USERBOOKS_JSON_PATH = "/Users/athipants/AthisProject/PythonServer/userBooks.json"

app.config['SESSION_TYPE'] = 'filesystem'
app.config['SESSION_FILE_DIR'] = './sessions'
app.config['SESSION_COOKIE_SECURE'] = False
app.config['SESSION_COOKIE_HTTPONLY'] = False
app.config['SESSION_COOKIE_SAMESITE'] = 'Lax'  # or 'Lax' or 'Strict'

Session(app)
```

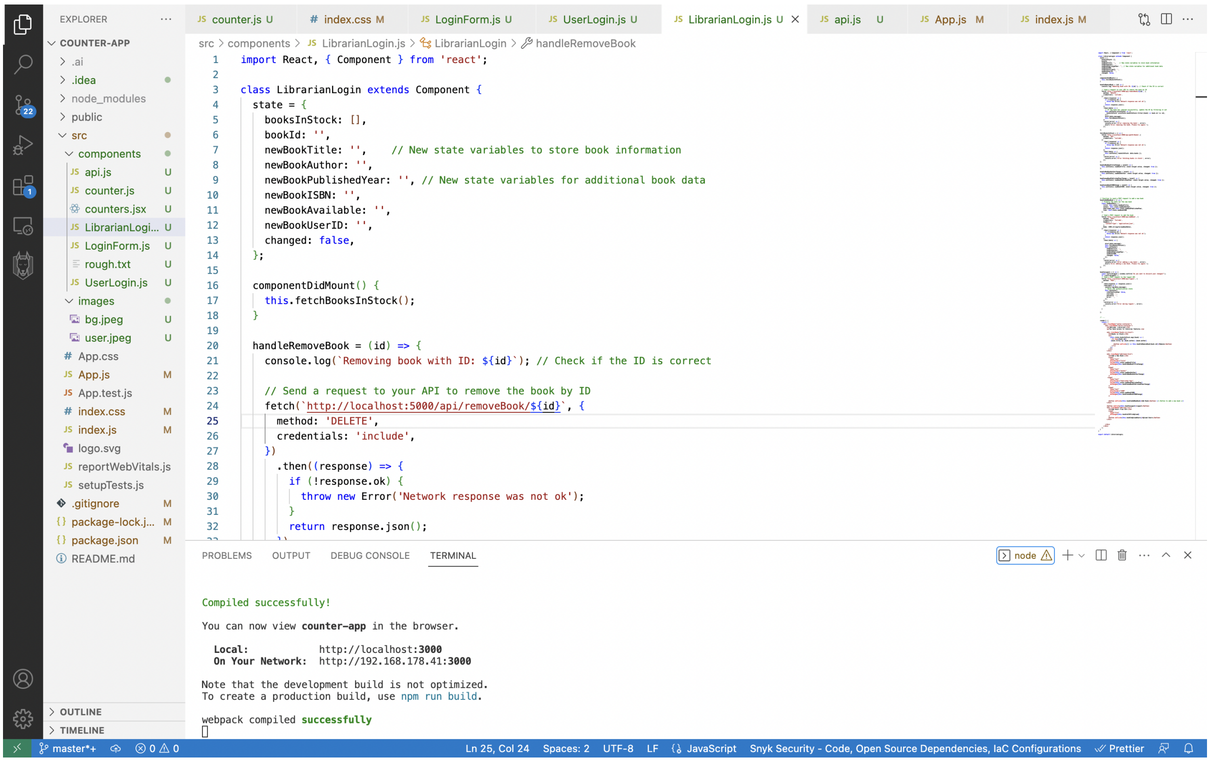
```
virtualenv          20.19.0
Werkzeug            2.2.3
wheel               0.33.1
zipp                1.2.0

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: /Library/Developer/CommandLineTools/usr/bin/python3 -m pip install --upgrade pip
⊗ (base) athipants@athipans-Air PythonServer % pip show
WARNING: ERROR: Please provide a package name or names.
⊗ (base) athipants@athipans-Air PythonServer % pip show all
WARNING: Package(s) not found: all
○ (base) athipants@athipans-Air PythonServer % █
```

Figure 5: Flask Application code

5 Installation of Front-end Dependencies for React JS Application

Setting up the front-end environment for a React JS application involves installing several key dependencies.

- **Node.js and npm:**
  - Node.js is a JavaScript runtime environment that is necessary to run JavaScript code outside a web browser nod (n.d.). npm (Node Package Manager) is included with Node.js and is used to manage JavaScript packages npm (n.d.).
  - Installation: Download and install Node.js from the official website.
  - npm will be installed automatically with Node.js.
- **React and React-DOM:**
  - React is a JavaScript library for building user interfaces, and React-DOM provides DOM-specific methods used in a web application rea (n.d.).
  - Installation: Typically, these are included when creating a new React app using Create React App. If manually installing, use `npm install react react-dom`.



```
import React, { Component } from 'react';

class LibrarianLogin extends Component {
  state = {
    booksInStock: [],
    bookId: '',
    newBookTitle: '',    // New state variables to store book information
    newBookAuthor: '',
    newBookPublishedYear: '', // New state variables for additional book data
    newBookISBN: '',
    newBookAvailable: '',
    newBookUserID: '',
    changed: false,
  };

  componentDidMount() {
    this.fetchBooksInStock();
  }

  handleRemoveBook = (id) => {
    console.log(`Removing book with ID: ${id}`); // Check if the ID is correct

    // Send a request to your API to remove the book by ID
    fetch(`http://localhost:5000/api/removeBook/${id}`, {
      method: 'DELETE',
      credentials: 'include',
    })
      .then((response) => {
        if (!response.ok) {
          throw new Error('Network response was not ok');
        }
        return response.json();
```

```
Compiled successfully!

You can now view counter-app in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.178.41:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Figure 6: React JS code

6 Installation of Scikit-learn and Training with Dataset

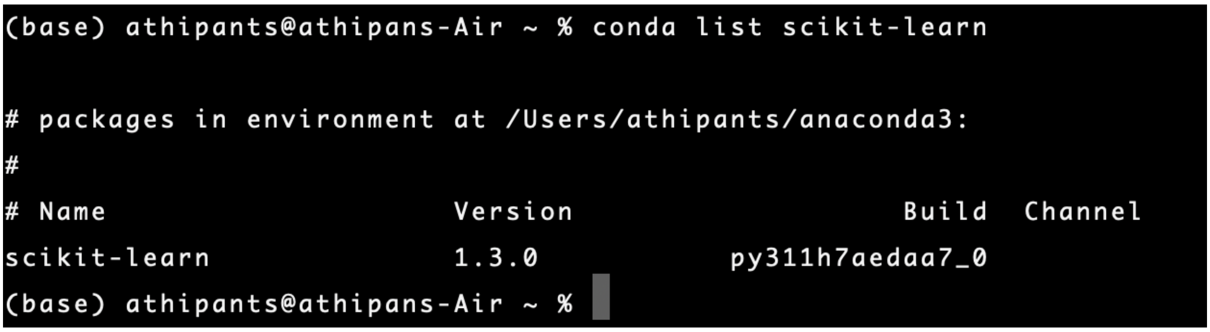
The Scikit-learn library is widely used for machine learning tasks in Python. It provides efficient tools for data analysis and is built on NumPy, SciPy, and Matplotlib. Below are the steps for installing Scikit-learn, training a machine learning model with a dataset, and integrating the trained model into a Flask application sci (n.d.).

Installation of Scikit-learn

- We dont need to install Scikit-learn package separately, as conda itself installs the package into our machine.

Training with Dataset

- **Data Loading and Preprocessing:**
  - Load your dataset using Pandas
- **Feature Extraction:**
  - Use TF-IDF for converting the string into numerical vectors.
- **Model Selection and Training:**



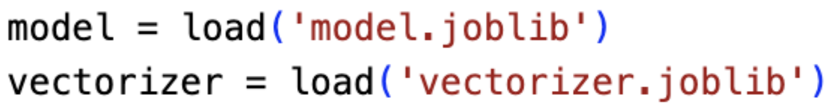
```
(base) athipants@athipans-Air ~ % conda list scikit-learn

# packages in environment at /Users/athipants/anaconda3:
#
# Name                    Version                   Build  Channel
scikit-learn              1.3.0           py311h7aedaa7_0
(base) athipants@athipans-Air ~ %
```

Figure 7: Scikit-learn installation

  - Use RandomForestClassifier for classification tasks.
  - Split the data into training and testing sets using train_test_split.
  - Train the model using the training data and evaluate its performance on the test set.

Integration with Flask Application

- **Export the Trained Model:**
  - After training, export the model using joblib: `joblib.dump(model, 'model.joblib')`.
- **Integrate Model into Flask:**
  - In your Flask application, import the trained model: `model = joblib.load('model.joblib`
  - Now, the model is ready to be used job (n.d.).



```
model = load('model.joblib')
vectorizer = load('vectorizer.joblib')
```

Figure 8: Joblib integration

7 Installation of PostgreSQL

PostgreSQL is a popular relational database management system which is used for the playground application.

7.0.1 Download PostgreSQL

The suitable file based on the chosen environment, need to be downloaded from the official site pos (n.d.). https://www.postgresql.org/download/

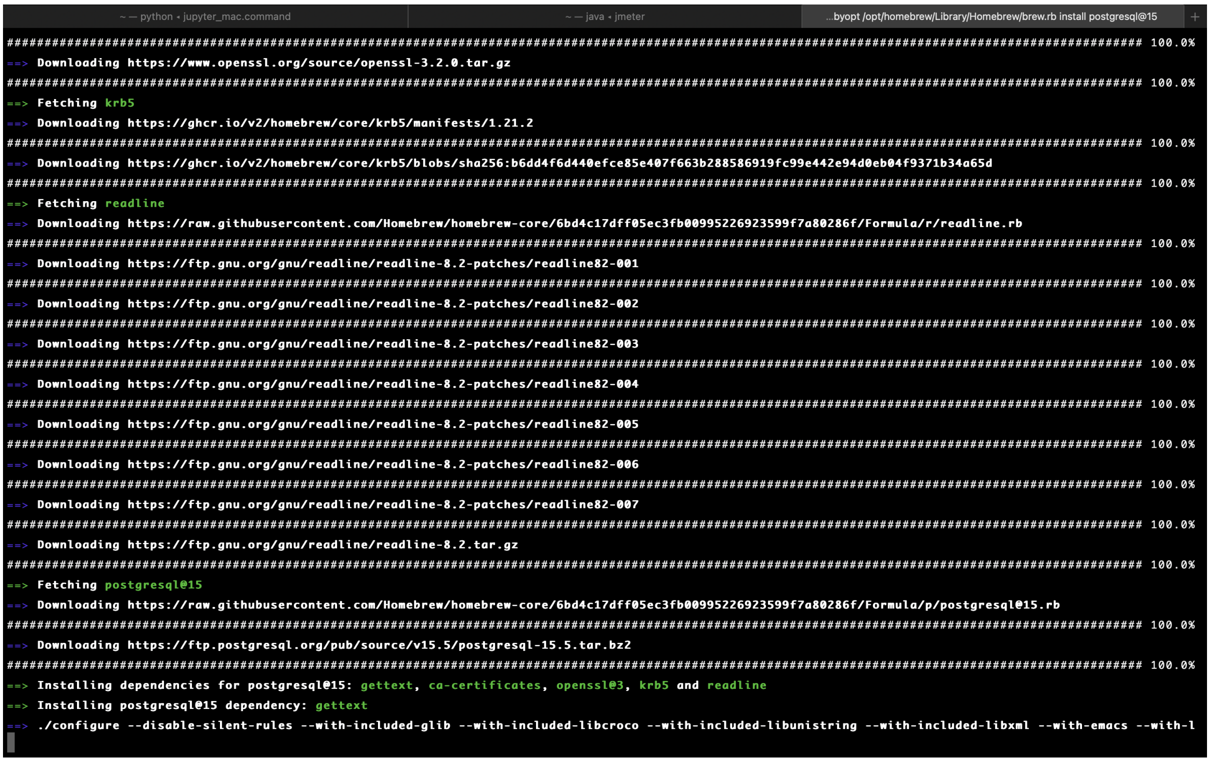
7.0.2 Select Version

Choose the appropriate version for your operating system. Click the download link for the chosen OS.

For macOS users, you can use Homebrew for a straightforward installation. Open the terminal and execute the following command:

brew install postgresql@15

This command can be used in the terminal to install the psql, as shown in the fig. **??**



~ — python ◂ jupyter_mac.command
~ — java ◂ jmeter
...byopt /opt/homebrew/Library/Homebrew/brew.rb install postgresql@15
+

```
########################################################################################################################################################## 100.0%
==> Downloading https://www.openssl.org/source/openssl-3.2.0.tar.gz
########################################################################################################################################################## 100.0%
==> Fetching krb5
==> Downloading https://ghcr.io/v2/homebrew/core/krb5/manifests/1.21.2
########################################################################################################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/krb5/blobs/sha256:b6dd4f6d440efce85e407f663b288586919fc99e442e94d0eb04f9371b34a65d
########################################################################################################################################################## 100.0%
==> Fetching readline
==> Downloading https://raw.githubusercontent.com/Homebrew/homebrew-core/6bd4c17dff05ec3fb00995226923599f7a80286f/Formula/r/readline.rb
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-001
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-002
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-003
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-004
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-005
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-006
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2-patches/readline82-007
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.gnu.org/gnu/readline/readline-8.2.tar.gz
########################################################################################################################################################## 100.0%
==> Fetching postgresql@15
==> Downloading https://raw.githubusercontent.com/Homebrew/homebrew-core/6bd4c17dff05ec3fb00995226923599f7a80286f/Formula/p/postgresql@15.rb
########################################################################################################################################################## 100.0%
==> Downloading https://ftp.postgresql.org/pub/source/v15.5/postgresql-15.5.tar.bz2
########################################################################################################################################################## 100.0%
==> Installing dependencies for postgresql@15: gettext, ca-certificates, openssl@3, krb5 and readline
==> Installing postgresql@15 dependency: gettext
==> ./configure --disable-silent-rules --with-included-glib --with-included-libcroco --with-included-libunistring --with-included-libxml --with-emacs --with-l
```

Figure 9: PostgreSQL download

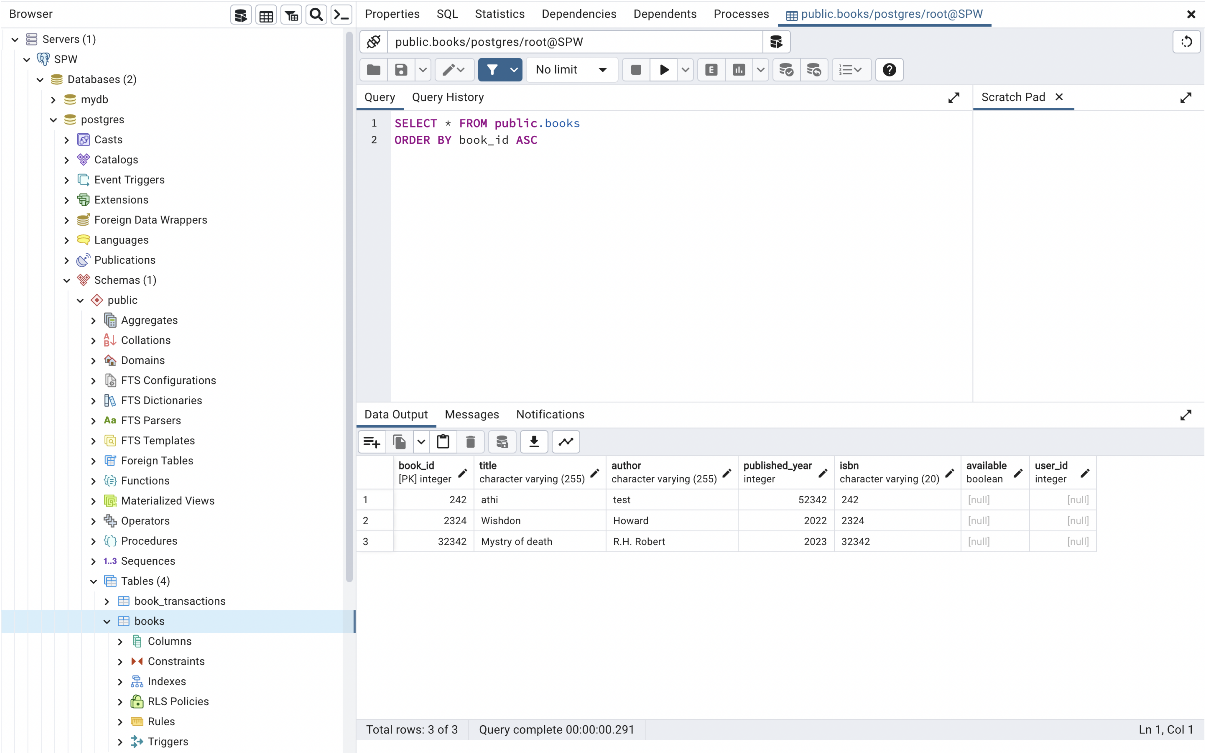
7.0.3 Installation of PG Admin

For a user-friendly database management interface, install PG Admin. Visit their official site:

`https://www.pgadmin.org/download/pgadmin-4-macos/`

Download the appropriate dmg or exe file, install it on your system, and launch PG Admin. Figure 10 provides a preview of PG Admin.

Fig. 10 shows the preview of PG Admin.



```
SELECT * FROM public.books
ORDER BY book_id ASC
```

	book_id [PK] integer	title character varying (255)	author character varying (255)	published_year integer	isbn character varying (20)	available boolean	user_id integer
1	242	athi	test	52342	242	[null]	[null]
2	2324	Wishdon	Howard	2022	2324	[null]	[null]
3	32342	Mystry of death	R.H. Robert	2023	32342	[null]	[null]

Total rows: 3 of 3 Query complete 00:00:00.291 Ln 1, Col 1

Figure 10: A view of PG admin

References

fla (n.d.).
URL: *https://flask.palletsprojects.com/*

job (n.d.).
URL: *https://joblib.readthedocs.io/*

nod (n.d.).
URL: *https://nodejs.org/en*

npm (n.d.).
URL: *https://docs.npmjs.com/*

pan (n.d.).
URL: *https://pandas.pydata.org/docs/*

pos (n.d.).
URL: *https://www.postgresql.org/*

pyt (n.d.).
URL: *https://www.python.org/*

rea (n.d.).
URL: *https://legacy.reactjs.org/*

sci (n.d.).
URL: *https://scikit-learn.org/*