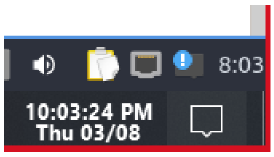
System Installation

Our system has been tested to be set up and run on VMware Player for Windows.

It is assumed that VMware Player for Windows has been downloaded and installed on a Windows machine, and that the Lubuntu image has been loaded into VMware. It is also assumed that within Lubuntu, the codebase has been downloaded and unzipped into a directory (whether by git cloning or downloading the codebase as a zip file). This installation was not tested successfully on VMware Fusion (for Mac OS).

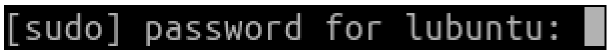
General notes during installation:
Timezone mismatch:
The default Lubuntu timezone does not match Australian Eastern Standard Time.



Since this project is related to event management, and the correct start/end times for an event is important, please be aware to use the Lubuntu system time as reference, and not the local machine system time (that VMware is running on).

Sudo password:
If prompted for [sudo] password for lubuntu:

Terminal prompt:



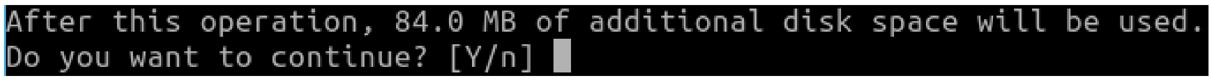
```
[sudo] password for lubuntu:
```

Enter:

lubuntu

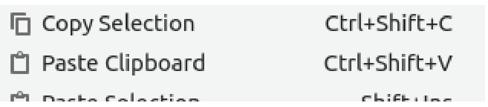
Do you want to continue?
During certain installations, do not forget to enter **Y** when prompted.

Terminal prompt:



```
After this operation, 84.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Paste from clipboard method:
The best method to paste a copied block of code into QTerminal, is by either right-clicking the terminal and selecting **Paste Clipboard**:

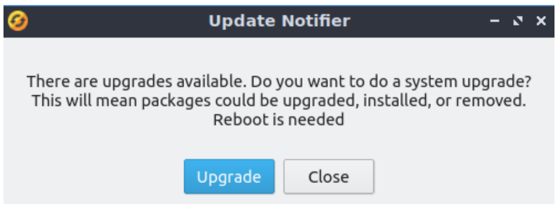


Copy Selection Ctrl+Shift+C
Paste Clipboard Ctrl+Shift+V
Paste Selection Shift+Ins

Or by using the **keyboard shortcut**:

```
Ctrl + Shift + V
```

Update Notifier:
If at any time a prompt appears asking if we would like to do a system upgrade, always click **Close**.



Update Notifier

There are upgrades available. Do you want to do a system upgrade?
This will mean packages could be upgraded, installed, or removed.
Reboot is needed

(Reason: Some sudo apt upgrades override the dependencies version we are relying on for the project.)

Screen Record of the Installation Process:
Here is a complementary YouTube video showing the entire process:
https://youtu.be/9086YuHwkzI

Steps 1:
Open a new terminal, and enter the following commands, one at a time, to install PostgreSQL:

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" >
/etc/apt/sources.list.d/pgdg.list'
```

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

```
sudo apt-get update
```

```
sudo apt-get -y install postgresql
```

Steps 2:
Now, we need to give superuser rights to the current system profile to enter the database:

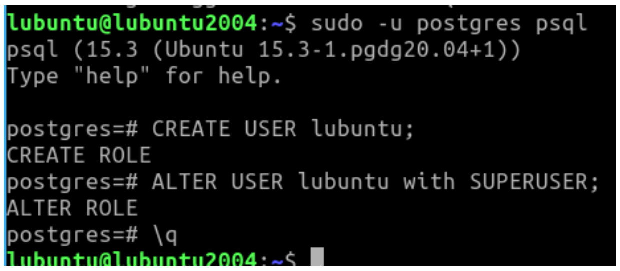
```
sudo -u postgres psql
```

```
CREATE USER lubuntu;
```

```
ALTER USER lubuntu WITH SUPERUSER;
```

```
\q
```

Terminal output:



```
lubuntu@lubuntu2004:~$ sudo -u postgres psql
psql (15.3 (Ubuntu 15.3-1.pgdg20.04+1))
Type "help" for help.

postgres=# CREATE USER lubuntu;
CREATE ROLE
postgres=# ALTER USER lubuntu with SUPERUSER;
ALTER ROLE
postgres=# \q
lubuntu@lubuntu2004:~$
```

Steps 3:

Now, let's install the Python package installer, pip, and use it to install other Python packages:

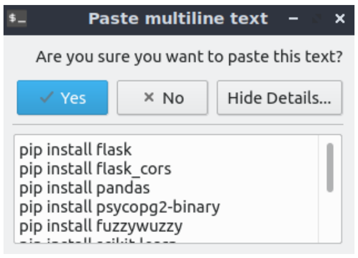
```
sudo apt install python3-pip
```

The following commands can be copied and pasted as one block. Note, the **newline** after the last pip install should be included in the copy:

```
pip install flask
pip install flask_cors
pip install pandas
pip install psycopg2-binary
pip install fuzzywuzzy
pip install scikit-learn
pip install meteostat
pip install xgboost
pip install python-Levenshtein

```

A prompt will appear, press **Yes**.

Terminal prompt:



Are you sure you want to paste this text?

```
pip install flask
pip install flask_cors
pip install pandas
pip install psycopg2-binary
pip install fuzzywuzzy
```

Steps 4:

Let us now install npm and nvm, the package manager and package updater for Javascript:

```
sudo apt install npm
```

```
sudo apt install curl
```

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.4/install.sh | bash
```

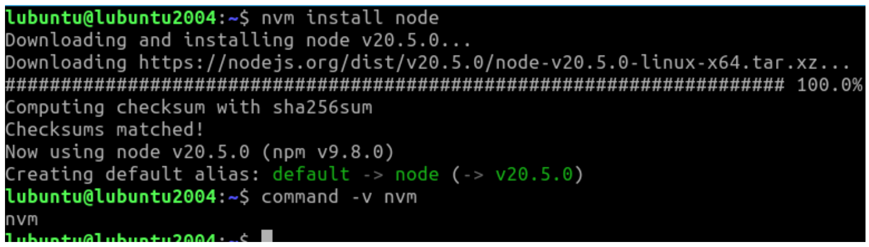
Important: Close the current terminal, and open a new terminal. Then continue:
(Reason: NVM only becomes active on new instances of the terminal after installation.)

```
nvm install node
```

```
command -v nvm
```

If all prior installation steps were successful, running *command -v nvm* should output **nvm** to the terminal.

Terminal output:



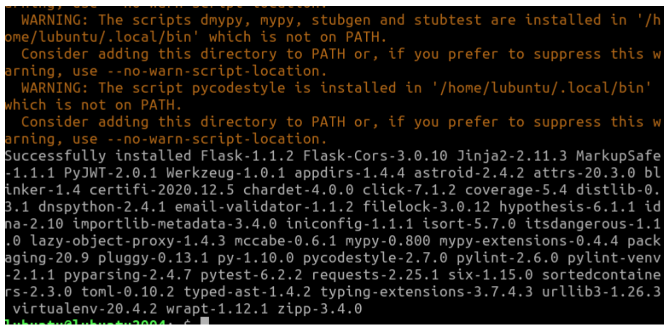
```
lubuntu@lubuntu2004:~$ nvm install node
Downloading and installing node v20.5.0...
Downloading https://nodejs.org/dist/v20.5.0/node-v20.5.0-linux-x64.tar.xz...
######################################################################## 100.0%
Computing checksum with sha256sum
Checksums matched!
Now using node v20.5.0 (npm v9.8.0)
Creating default alias: default -> node (-> v20.5.0)
lubuntu@lubuntu2004:~$ command -v nvm
nvm
```

```
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash - && sudo apt-get install -y nodejs
```

```
pip3 install $(curl https://www.cse.unsw.edu.au/~cs1531/21T3/requirements.txt)
```

These warnings are expected. Ignore, and continue.

Terminal output:



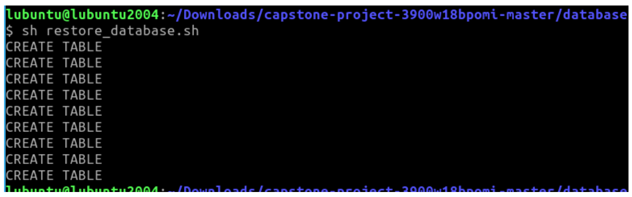
```
  WARNING: The scripts dmypy, mypy, stubgen and stubtest are installed in '/h
ome/lubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this w
arning, use --no-warn-script-location.
  WARNING: The script pycodestyle is installed in '/home/lubuntu/.local/bin'
which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this w
arning, use --no-warn-script-location.
Successfully installed Flask-1.1.2 Flask-Cors-3.0.10 Jinja2-2.11.3 MarkupSafe
-1.1.1 PyJWT-2.0.1 Werkzeug-1.0.1 appdirs-1.4.4 astroid-2.4.2 attrs-20.3.0 bl
inker-1.4 certifi-2020.12.5 chardet-4.0.0 click-7.1.2 coverage-5.4 distlib-0.
3.1 dnspython-2.4.1 email-validator-1.1.2 filelock-3.0.12 hypothesis-6.1.1 id
na-2.10 importlib-metadata-3.4.0 iniconfig-1.1.1 isort-5.7.0 itsdangerous-1.1
.0 lazy-object-proxy-1.4.3 mccabe-0.6.1 mypy-0.800 mypy-extensions-0.4.4 pack
aging-20.9 pluggy-0.13.1 py-1.10.0 pycodestyle-2.7.0 pylint-2.6.0 pylint-venv
-2.1.1 pyparsing-2.4.7 pytest-6.2.2 requests-2.25.1 six-1.15.0 sortedcontaine
rs-2.3.0 toml-0.10.2 typed-ast-1.4.2 typing-extensions-3.7.4.3 urllib3-1.26.3
 virtualenv-20.4.2 wrapt-1.12.1 zipp-3.4.0
```

Step 5:
Important: Restart the Lubuntu virtual machine.

Step 6:
After restarting, open a terminal and point to the **database directory** of the project (*capstone-project-3900w18bpomi/database*). If PostgreSQL was installed successfully, running the following command will successfully create the tables needed for the project:

```
sh restore_database.sh
```

Terminal output:



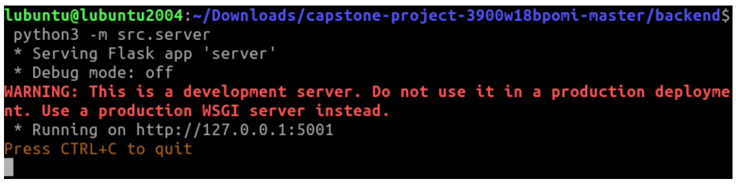
```
lubuntu@lubuntu2004:~/Downloads/capstone-project-3900w18bpomi-master/database
$ sh restore_database.sh
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
lubuntu@lubuntu2004:~/Downloads/capstone-project-3900w18bpomi-master/database
```

Step 7:
Let us now start the server. Point the current terminal to the **backend directory** of the project (*capstone-project-3900w18bpomi/backend*) and enter:

```
python3 -m src.server
```

If all prior installation steps were successful, the flask server should now be active with no errors:

Terminal output:



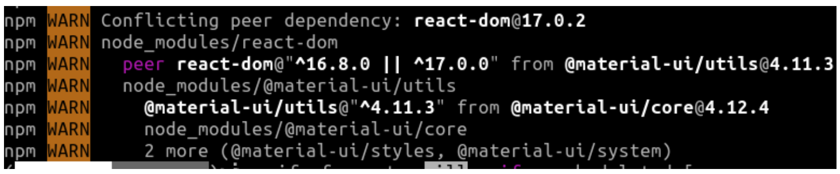
```
lubuntu@lubuntu2004:~/Downloads/capstone-project-3900w18bpomi-master/backend$
 python3 -m src.server
 * Serving Flask app 'server'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployme
nt. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5001
Press CTRL+C to quit
```

Steps 8:
Now, let us build the React app used for the frontend, and start the frontend.

Important: Open a **new terminal** (i.e., leave the server terminal running) point this terminal to the **frontend directory** of the project (*capstone-project-3900w18bpomi/frontend*). Run:

```
npm install --force
```

Terminal output:



```
npm WARN Conflicting peer dependency: react-dom@17.0.2
npm WARN node_modules/react-dom
npm WARN   peer react-dom@"^16.8.0 || ^17.0.0" from @material-ui/utils@4.11.3
npm WARN   node_modules/@material-ui/utils
npm WARN     @material-ui/utils@"^4.11.3" from @material-ui/core@4.12.4
npm WARN     node_modules/@material-ui/core
npm WARN     2 more (@material-ui/styles, @material-ui/system)
```

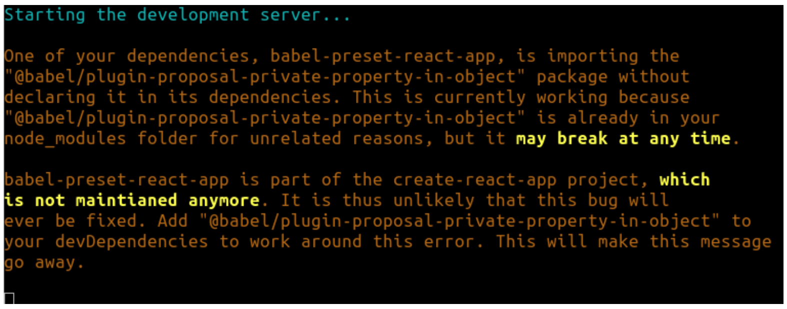
(This step may take awhile.)

```
npm audit fix
```

```
npm start
```

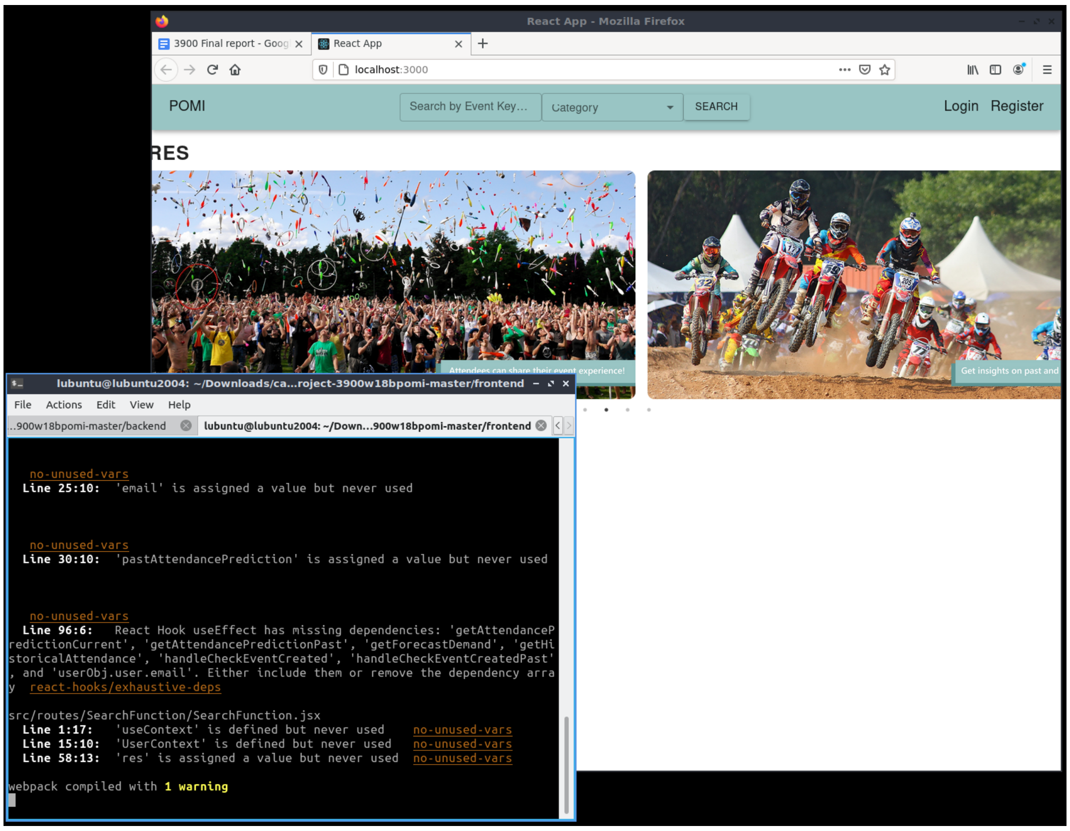
When running *npm start*, there may be warnings. These are expected. Ignore and continue.

Terminal output:



```
Starting the development server...

One of your dependencies, babel-preset-react-app, is importing the
"@babel/plugin-proposal-private-property-in-object" package without
declaring it in its dependencies. This is currently working because
"@babel/plugin-proposal-private-property-in-object" is already in your
node_modules folder for unrelated reasons, but it may break at any time.

babel-preset-react-app is part of the create-react-app project, which
is not maintianed anymore. It is thus unlikely that this bug will
ever be fixed. Add "@babel/plugin-proposal-private-property-in-object" to
your devDependencies to work around this error. This will make this message
go away.
```

We should now have a terminal running the server, another terminal running the React app, and an open webpage to the landing page of our project.

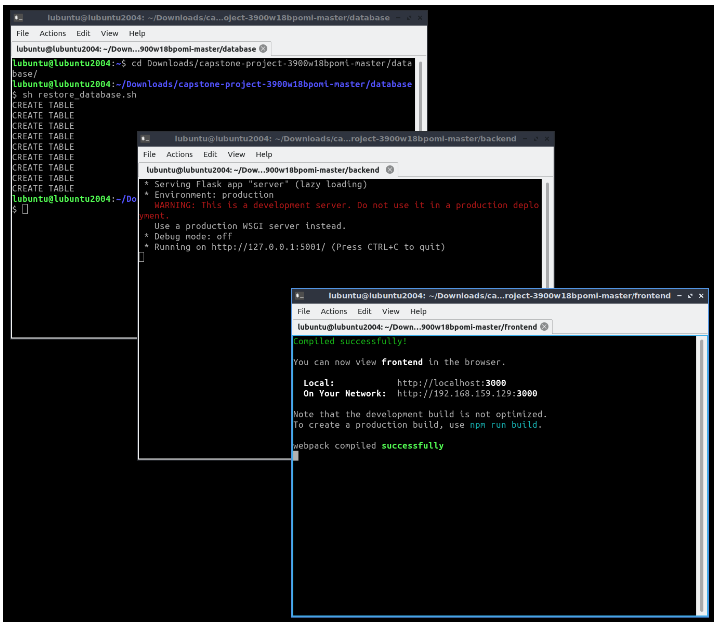

React App - Mozilla Firefox
3900 Final report - Goog
React App
localhost:3000
POMI
Search by Event Key...
Category
SEARCH
Login
Register
RES
Attendees can share their event experience!
Get insights on past and
lubuntu@lubuntu2004: ~/Downloads/ca...roject-3900w18bpomi-master/frontend
File
Actions
Edit
View
Help
..900w18bpomi-master/backend
lubuntu@lubuntu2004: ~/Down...900w18bpomi-master/frontend

```
  no-unused-vars
  Line 25:10:  'email' is assigned a value but never used


  no-unused-vars
  Line 30:10:  'pastAttendancePrediction' is assigned a value but never used


  no-unused-vars
  Line 96:6:   React Hook useEffect has missing dependencies: 'getAttendanceP
redictionCurrent', 'getAttendancePredictionPast', 'getForecastDemand', 'getHi
storicalAttendance', 'handleCheckEventCreated', 'handleCheckEventCreatedPast'
, and 'userObj.user.email'. Either include them or remove the dependency arra
y  react-hooks/exhaustive-deps

src/routes/SearchFunction/SearchFunction.jsx
  Line 1:17:   'useContext' is defined but never used    no-unused-vars
  Line 15:10:  'UserContext' is defined but never used   no-unused-vars
  Line 58:13:  'res' is assigned a value but never used  no-unused-vars

webpack compiled with 1 warning
```

System Documentation

System usage:

In practice, we usually have three terminals open. One each pointing to the database, backend, and frontend folders respectively.



lubuntu@lubuntu2004: ~/Downloads/ca...oject-3900w18bpomi-master/database

```
lubuntu@lubuntu2004:~$ cd Downloads/capstone-project-3900w18bpomi-master/data
base/
lubuntu@lubuntu2004:~/Downloads/capstone-project-3900w18bpomi-master/database
$ sh restore_database.sh
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
lubuntu@lubuntu2004:~/Do
$ 
```

lubuntu@lubuntu2004: ~/Downloads/ca...roject-3900w18bpomi-master/backend

```
 * Serving Flask app "server" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deplo
yment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5001/ (Press CTRL+C to quit)
```

lubuntu@lubuntu2004: ~/Downloads/ca...roject-3900w18bpomi-master/frontend

```
Compiled successfully!

You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.159.129:3000

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

webpack compiled successfully
```

Database usage:

To restore the database to a blank state (no registered accounts, no events etc.) we would run in the terminal pointing to the **database directory** (*capstone-project-3900w18bpomi/database*):

```
sh restore_database.sh
```

This is a shell script that deletes the existing database and re-creates it, and populates it with empty tables. If you would like to manually enter the database, in the database terminal, enter:

```
sudo -u postgres psql
```

Followed by:

```
\c COMP3900
```

Now you can view the data stored in the database manually, by entering commands such as, but not limited to:

```
select * from users;
```
… to list all registered users.

```
select * from events;
```
… to list all created events.

```
select * from historical events where creator = ‘email@mail.com’;
```
… to list all events that have passed that were created by a particular user (email).

Note: The database cannot be restored if the server is running.

Server usage:
To run the server, we would point another terminal to the **backend directory** (*capstone-project-3900w18bpomi/backend*) and run:

```
python3 -m src.server
```

Note 1: To restore the database, the server must first be shut down.
Note 2: To test persistence, only the server needs to be shut down. The React app for the frontend does not need to be shut down.

Frontend usage:
To run the React app that powers the frontend, point the last terminal to the **frontend directory** (*capstone-project-3900w18bpomi/frontend*) and run:

```
npm start
```

Note: The server needs to be started first before the frontend runs.