IT2002 – Project Tutorial

A minimalistic web application for Database Management

Prerequisites for the project

- Basic knowledge of HTML, CSS and JavaScript
- Intermediate knowledge to program in Python
- Ability to use SSH on bash or powershell to operate the virtual server
- Preliminary understanding of servers and webpage deployment
- To be able to work in PostgreSQL and its UI clients

Prototype Demo

Accessing Virtual Machine

- SoC IT Office will provide individual groups with VMs
- There might still be need for port access activation (e.g. 8080)
- Additional VM and port access activation can be done at: https://rt.comp.nus.edu.sg/
- Provided VMs should have necessary packages already installed
- Your SSH access URL will be as sadm@<some-name>.comp.nus.edu.sg
- Use `ssh sadm@<some-name>.comp.nus.edu.sg` and enter your root password

External access and VPN

- You cannot access the virtual machines outside SoC network
- Hence, you will need a VPN application (FortiClient as suggested)
- You can find detailed info at: https://dochub.comp.nus.edu.sg/cf/guides/network/vpn
- You will need your NUSNET ID and password to log into the VPN service
- As said and experimented, port 80 should automatically be enabled to interact with HTTP requests
- Refer to the request page in case you face problems with it

VSCode to better interact with VM

- Visual Studio Code allows you to remotely develop and deploy on the cloud and distant servers
- Install it from the official link: https://code.visualstudio.com/download
- Then go to the command palette using Ctrl (Cmd) + Shift + P
- Select ”+ Add new host” and type sadm@<your-given-id>.comp.nus.edu.sg
- Continue with you admin password and have an access to file system, terminal, and many more

Synchronizing your work with git

- Refer to git’s official website to set it up on your PC: https://git-scm.com/downloads
- Create a repository on GitHub and share it with your teammates
- You will need to enable add both VM’s and your individual PC’s public ssh keys to GitHub
- Use it to simultaneously update the code on the server
- Prefer to develop the flow on your machines, merge your parts in the end

Postgres database visualization

- Conveniently, you will need pgAdmin installed on your machine
- Postgres should be already running on your virtual machine (server)
- Enter your virtual server address (given by SoC), username and password both as “postgres”
- There should already exist a database called “postgres”
- Alternatively, your servers have adminer pre-installed
- Type your ip-address/adminer in browser to access it

Running your code

- Project architecture allows you to focus on the database management part
- A simple `python3 app.py` command will suffice to run the app on the server
- Everything visual is stored in the `view` folder
- Redirect to that folder and run `npm run dev` command, which will give you the host and port
- Give `npm run dev -- --host` flag if you are on the server and cannot access it from your local machine

Debugging your code

- VSCode is a free and open-source editor supporting all programming languages in this project
- Make sure you have the Python extension by Microsoft installed in extensions menu
- To debug your main app, simply press F5 key when it is open
- For the view application, press the Debug key in package.json file
- Use Postman or Insomnia to focus on the Python and PostgreSQL side of the project

Deploying your code

- No specific server and reverse-proxy configuration needed
- We’ll use background sessions with tmux
- Use `tmux new-session –s <session-name>` to initialize it
- Run `python3 app.py` and detach from the session for the database app
- Run `npm run build` and `npm run preview` in the other session and detach
- Alternatively, you can run `waitress-serve --port=2222 --call app:create app` for the database app