Complete SRE environment with WSL, Minikube, Prometheus, Grafana, and our React-based dashboard application

1. WSL not detected:

```
hafsa_027@Dell:~/Grafana/proj2$ ./complete-react-sre-script.sh
[INFO] 2025-03-20 05:26:30 - Starting React SRE Application setup
[INFO] 2025-03-20 05:26:30 - Checking WSL environment...
[ERROR] 2025-03-20 05:26:30 - This script must be run within WSL
```

```
hafsa_027@Dell:~/Grafana/proj2$ vi complete-react-sre-script.sh
hafsa_027@Dell:~/Grafana/proj2$ ./complete-react-sre-script.sh
[INFO] 2025-03-20 05:41:49 - Starting React SRE Application setup
[INFO] 2025-03-20 05:41:49 - Checking WSL environment...
[INFO] 2025-03-20 05:41:49 - WSL detected, continuing with setup
[INFO] 2025-03-20 05:41:49 - Creating project directory structure
[INFO] 2025-03-20 05:41:49 - Project directories created at /home/hafsa_027/react-sre-project
[INFO] 2025-03-20 05:41:49 - Installing system dependencies...
[INFO] 2025-03-20 05:41:49 - Updating package lists
[sudo] password for hafsa_027:
```

Replace with:- **grep -qEi "(microsoft|wsl)" /proc/version**

-q → Quiet mode: The script only displays if WSL is detected (exit status 0 for match, 1 for no match).
-E → Extended Regex: For the | (OR) operator, matching either "microsoft" or "wsl".
-i → Ensures matches are case-insensitive (WSL, wsl, Microsoft, MICROSOFT all match).

2. Latest version of grafana does not support React 19 so install React 18:

```
Happy hacking!
npm error code ERESOLVE
npm error ERESOLVE unable to resolve dependency tree
npm error
npm error While resolving: sre-react-app@0.1.0
npm error Found: react@19.0.0
npm error node_modules/react
npm error   react@"^19.0.0" from the root project
npm error
npm error Could not resolve dependency:
npm error peer react@"^18.0.0" from @grafana/ui@11.5.2
npm error node_modules/@grafana/ui
npm error   @grafana/ui@"*" from the root project
npm error
npm error Fix the upstream dependency conflict, or retry
npm error this command with --force or --legacy-peer-deps
npm error to accept an incorrect (and potentially broken) dependency resolution.
npm error
npm error
npm error For a full report see:
npm error /home/hafsa_027/.npm/_logs/2025-03-20T05_45_59_676Z-eresolve-report.txt
npm error A complete log of this run can be found in: /home/hafsa_027/.npm/_logs/2025-03-20T05_45_59_676Z-debug-0.log
```

Replace react 19 with react 18

```
# Manually install React 18 (to avoid conflicts
with @grafana/ui)
echo "[INFO] Downgrading React to version 18..."
npm install react@18 react-dom@18


  # Install additional dependencies for SRE
  npm install --save \
    react@18 react-dom@18 \
```

3. Module not found:

```
Traceback (most recent call last):
  File "/usr/lib/cnf-update-db", line 3, in <module>
    import apt_pkg
ModuleNotFoundError: No module named 'apt_pkg'
Reading package lists... Done
W: http://pkg.jenkins.io/debian-stable/binary/Release.gpg: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg)
, see the DEPRECATION section in apt-key(8) for details.
E: Problem executing scripts APT::Update::Post-Invoke-Success 'if /usr/bin/test -w /var/lib/command-not-found/ -a -e /usr/lib
/cnf-update-db; then /usr/lib/cnf-update-db > /dev/null; fi'
E: Sub-process returned an error code
[ERROR] 2025-03-20 05:51:05 - Failed to update package lists
```

apt_pkg exists but might not be correctly linked or found in your Python environment.

```
hafsa_027@Dell:~/Grafana/proj2$ python3 -V
Python 3.9.21
hafsa_027@Dell:~/Grafana/proj2$ ls -l /usr/lib/python3/dist-packages/ | grep apt_pkg
drwxr-xr-x  2 root root   4096 Mar 20 06:03 apt_pkg-stubs
-rw-r--r--  1 root root 347328 Jan 27 11:40 apt_pkg.cpython-312-x86_64-linux-gnu.so
```

```
hafsa_027@Dell:~/Grafana/proj2$ python3 --version
Python 3.9.21
hafsa_027@Dell:~/Grafana/proj2$ ls /usr/bin/python*
/usr/bin/python3  /usr/bin/python3-config  /usr/bin/python3.12  /usr/bin/python3.12-config  /usr/bin/python3.9  /usr/bin/pyth
on3.9-config
hafsa_027@Dell:~/Grafana/proj2$ sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.9 1
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.12 2
[sudo] password for hafsa_027:
update-alternatives: using /usr/bin/python3.12 to provide /usr/bin/python3 (python3) in auto mode
hafsa_027@Dell:~/Grafana/proj2$ sudo update-alternatives --config python3
There are 2 choices for the alternative python3 (providing /usr/bin/python3).

  Selection    Path                Priority   Status
------------------------------------------------------------
* 0            /usr/bin/python3.12   2         auto mode
  1            /usr/bin/python3.12   2         manual mode
  2            /usr/bin/python3.9    1         manual mode

Press <enter> to keep the current choice[*], or type selection number: 1
hafsa_027@Dell:~/Grafana/proj2$ python3 --version
Python 3.12.3
```

4. python setup.py develop did not run successfully.

pyproject.toml instead of setup.py

setup.py: a Python script traditionally used for defining a package.

toml: Defines package metadata in a structured, declarative format (TOML syntax). Replaces setup.py and setup.cfg by allowing different build systems (like setuptools, poetry, hatch).# Create a minimal pyproject.toml for editable installs

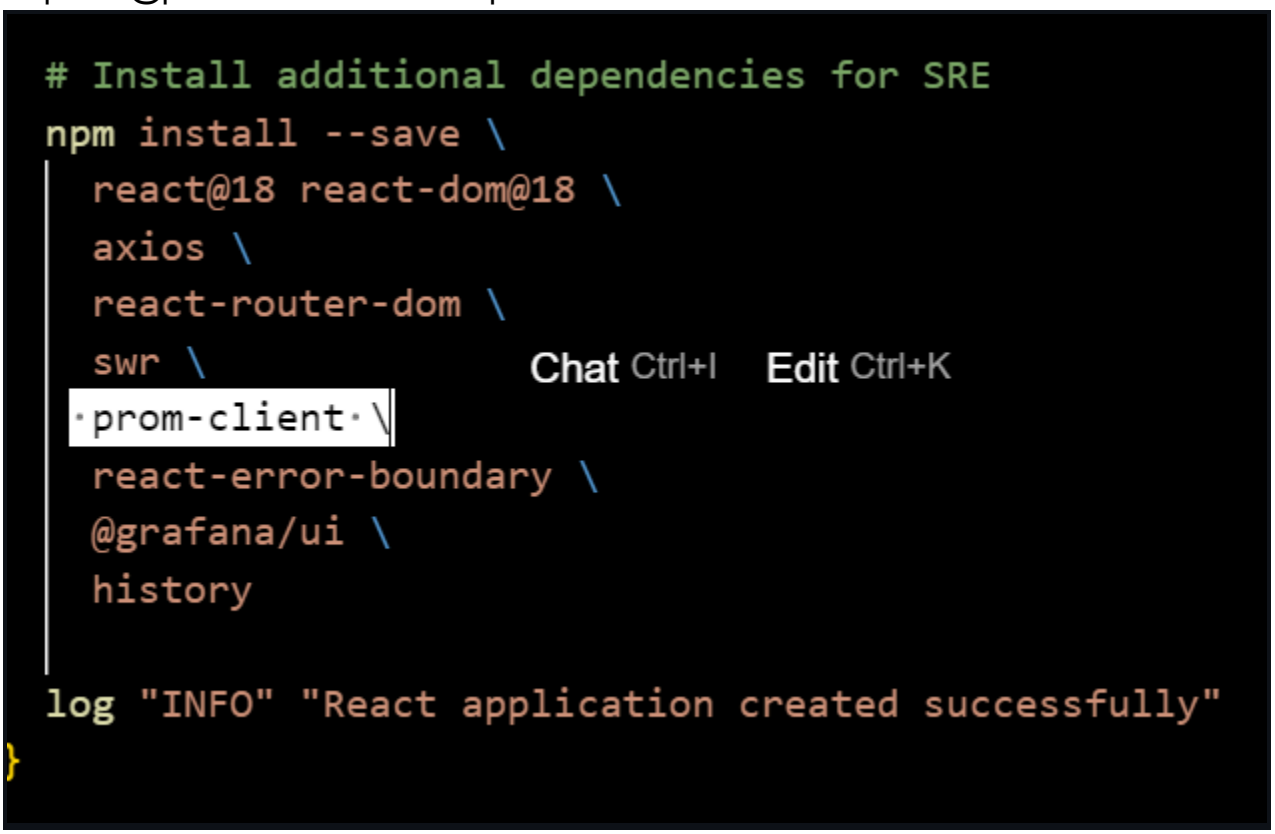
Replace the code with :

```
cat > "$PROJECT_ROOT/wzegh/pyproject.toml" << EOL
[build-system]
requires = ["setuptools", "wheel"]
build-backend = "setuptools.build_meta"

[project]
name = "wzegh"
version = "0.1.0"
description = "Mock implementation of wzegh library"
dependencies = []

EOL
# Install the mock wzegh library
cd "$PROJECT_ROOT/wzegh"
pip install -e .
```

5. @prometheus/client@ not found in registry:

```
Run `npm audit` for details.
npm warn ERESOLVE overriding peer dependency
npm error code E404
npm error 404 Not Found - GET https://registry.npmjs.org/@prometheus%2fclient - Not found
npm error 404
npm error 404  '@prometheus/client@*' is not in this registry.
npm error 404
npm error 404 Note that you can also install from a
npm error 404 tarball, folder, http url, or git url.
npm error A complete log of this run can be found in: /home/hafsa_027/.npm/_logs/2025-03-20T08_13_51_369Z-debug-0.log
```

Replace @prometheus/client with prom-client



6.

```
hafsa_027@Dell:~/Grafana/proj2$ cd /home/hafsa_027/react-sre-project
hafsa_027@Dell:~/react-sre-project$ ls
deploy_sre_app.sh  kubernetes  monitoring  requirements.txt  scripts  sre-react-app  venv  wzegh
hafsa_027@Dell:~/react-sre-project$ code .
hafsa_027@Dell:~/react-sre-project$ ./deploy_sre_app.sh
[INFO] 2025-03-20 09:38:15 - Starting React SRE Application deployment
[ERROR] 2025-03-20 09:38:15 - This script must be run within WSL
```

grep -qEi "(microsoft|wsl)" /proc/version

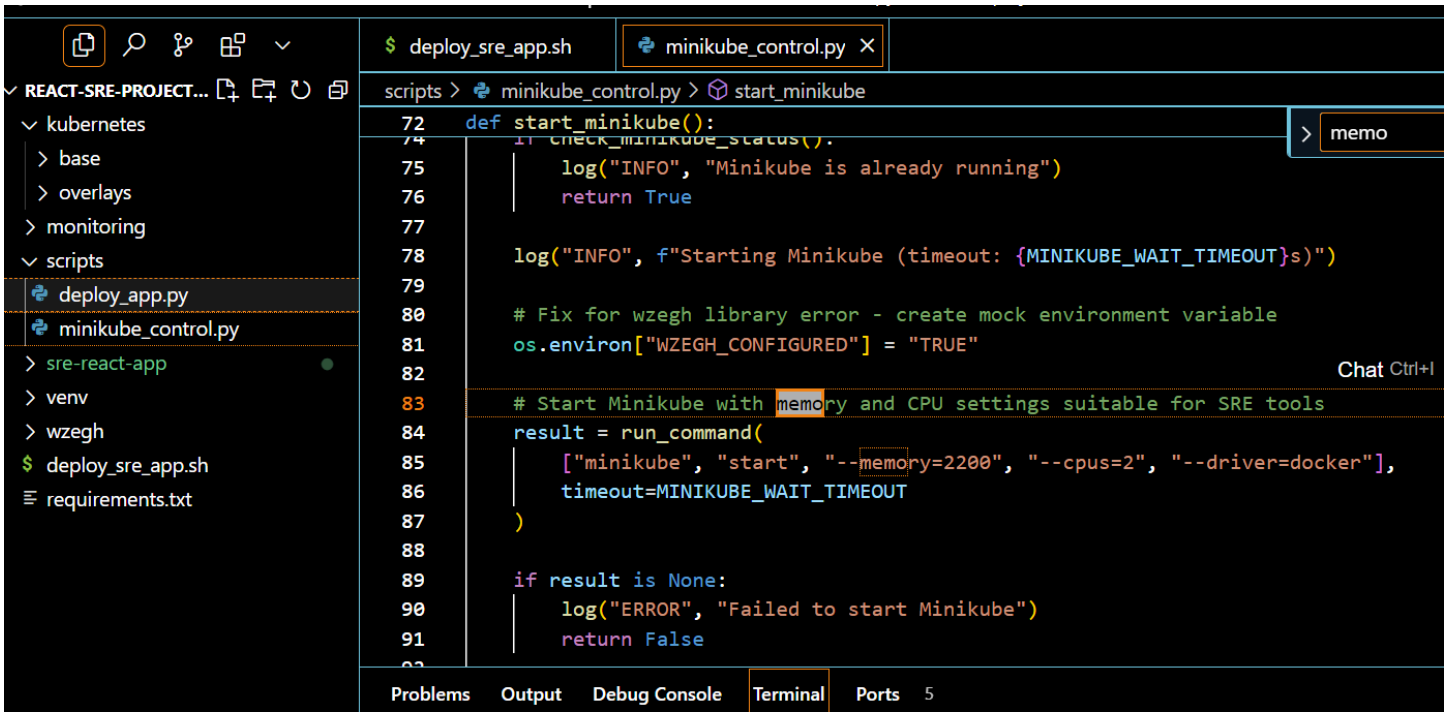
$ deploy_sre_app.sh

~/react-sre-project/deploy_sre_app.sh

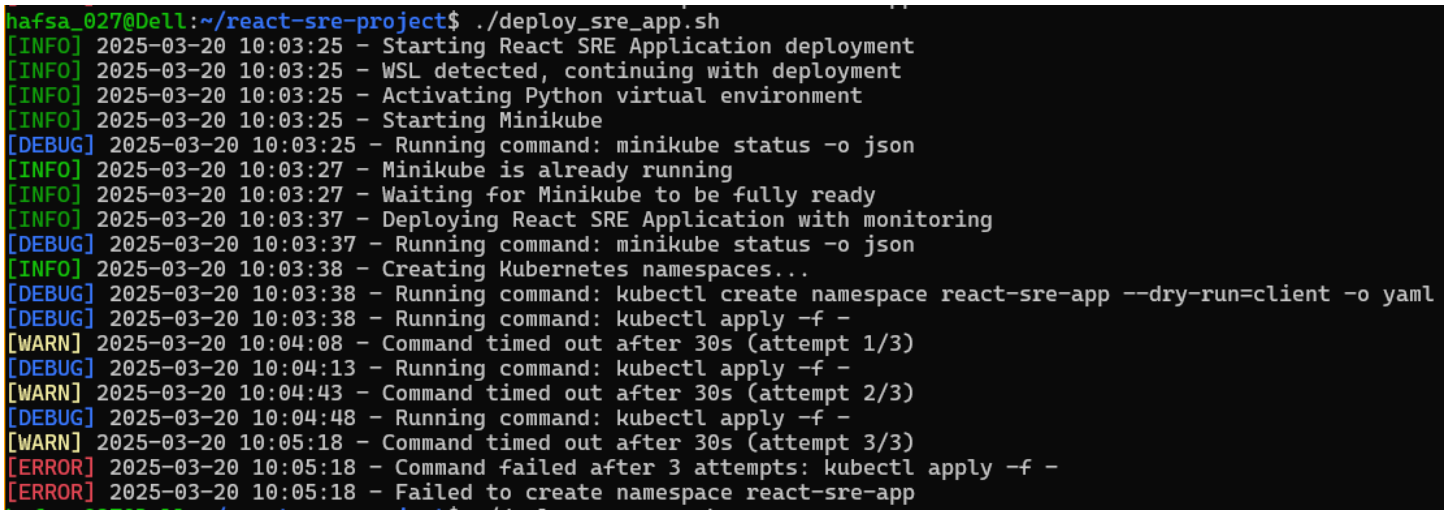
```
21  {
26      case $level in
39      esac
40  }
41
42  # Check if we're running in WSL
43  check_wsl() {
44      if ! grep -qEi "(microsoft|wsl)" /proc/version; then
45          log "ERROR" "This script must be run within WSL"
46          exit 1
47      fi
48
49      log "INFO" "WSL detected, continuing with deployment"
50  }
```

7. Allocation of memory:

```
[DEBUG] 2025-03-20 09:40:27 - Running command: minikube start --memory=4096 --cpus=2 --driver=docker
[WARN] 2025-03-20 09:40:28 - Command failed (attempt 2/3): X Exiting due to RSRC_OVER_ALLOC_MEM: Requested memory allocation 4096MB is more than your system limit 3802MB.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

[DEBUG] 2025-03-20 09:40:33 - Running command: minikube start --memory=4096 --cpus=2 --driver=docker
[WARN] 2025-03-20 09:40:34 - Command failed (attempt 3/3): X Exiting due to RSRC_OVER_ALLOC_MEM: Requested memory allocation 4096MB is more than your system limit 3802MB.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

[ERROR] 2025-03-20 09:40:39 - Command failed after 3 attempts: minikube start --memory=4096 --cpus=2 --driver=docker
[ERROR] 2025-03-20 09:40:39 - Failed to start Minikube
```



8. kubectl namespace is not created:



Replace:

```
apply_result = run_command(
    f"{KUBECTL_CMD} apply -f -",
    timeout=30,
    shell=True,
    retry=3,
)
```

With:

```
apply_result = run_command(
    f"{KUBECTL_CMD} create namespace {namespace} --dry-run=client -o yaml | {KUBECTL_CMD} apply -f -",
    timeout=30,
    shell=True,
    retry=3,
)
```

--dry-run=client -o yaml → Does a "dry run" of the command and outputs YAML instead of actually applying it.

- --dry-run=client: Simulates the command **without making changes**.
- -o yaml: Outputs the result in YAML format.

✅ This ensures the YAML is correctly piped into kubectl apply.

9. TypeError: run_command() got an unexpected keyword argument 'cwd'

```
[INFO] 2025-03-20 10:07:23 - Kubernetes namespaces created successfully
[INFO] 2025-03-20 10:07:23 - Building React application...
[INFO] 2025-03-20 10:07:23 - Installing npm dependencies...
Traceback (most recent call last):
  File "/home/hafsa_027/react-sre-project/scripts/deploy_app.py", line 385, in <module>
    success = main()
  File "/home/hafsa_027/react-sre-project/scripts/deploy_app.py", line 359, in main
    if not build_react_app():
  File "/home/hafsa_027/react-sre-project/scripts/deploy_app.py", line 99, in build_react_app
    result = run_command(["npm", "install"], timeout=300, cwd=REACT_APP_DIR)
TypeError: run_command() got an unexpected keyword argument 'cwd'
```

```
def run_command(command, timeout=60, retry=1, shell=False, cwd=None):
    """Run a shell command with timeout, retry logic, and optional working directory."""
    for attempt in range(retry):
        try:
            log("DEBUG", f"Running command: {command if shell else ' '.join(command)} in {cwd or 'current directory'}")

            if shell:
                result = subprocess.run(
                    command,
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                    text=True,
                    timeout=timeout,
                    shell=True,
                    cwd=cwd  # Add cwd here
                )
            else:
                result = subprocess.run(
                    command,
                    stdout=subprocess.PIPE,
                    stderr=subprocess.PIPE,
                    text=True,
                    timeout=timeout,
                    cwd=cwd  # Add cwd here
                )
```