Deployment Instructions for ECGScan

1. Launch the Instance and Update the System

After creating the server (e.g., with m1.medium flavor: 4GB RAM, 2 VCPU, 40GB disk), SSH into it and run the following:

```
sudo apt update
sudo apt upgrade
```

Install required packages like Python 3, pip, Nginx, Node.js, and NVM if they are not already present.

2. Clone the Repository and Switch Branch

Run the following to clone the codebase:

```
git clone git@github.com:UAlberta-CMPUT401/w25project-DepartmentofMedicine.git
cd w25project-DepartmentofMedicine
git checkout deployment_branch
```

3. Set Up the Backend

Navigate to the backend directory:

```
cd backend
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

Create the backend systemd service file at /etc/systemd/system/ecgscan-backend.service with the following content:

```
[Unit]
Description=ECGScan Flask Backend
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/w25project-DepartmentofMedicine/backend
ExecStart=/home/ubuntu/w25project-DepartmentofMedicine/backend/venv/bin/python3
/home/ubuntu/w25project-DepartmentofMedicine/backend/app.py
Restart=always

[Install]
```

WantedBy=multi-user.target

Enable and start the backend service:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable ecgscan-backend
sudo systemctl start ecgscan-backend
```

4. Set Up the Frontend

Navigate to the frontend directory:

```
cd ../frontend
npm install
```

Create the frontend systemd service file at /etc/systemd/system/frontend.service with the following content:

```
[Unit]
Description=ECGScan Frontend (React/Node)
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/w25project-DepartmentofMedicine/frontend
ExecStart=/bin/bash -c 'source /home/ubuntu/.nvm/nvm.sh && npm run dev -- --host 0.0.0.0'
Restart=always

[Install]
WantedBy=multi-user.target
```

Enable and start the frontend service:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl enable frontend
sudo systemctl start frontend
```

5. Configure Nginx

Edit the Nginx default site config at /etc/nginx/sites-available/default and paste the following:

```
server {
    listen [2605:fd00:4:1001:f816:3eff:fe72:24ea]:80;
    server_name 2605:fd00:4:1001:f816:3eff:fe72:24ea;
```

```
    location /api/ {
        proxy_pass http://127.0.0.1:5000;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    location / {
        proxy_pass http://10.2.14.55:5173/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

Then reload Nginx:

```
sudo nginx -t
sudo systemctl reload nginx
```

6. Security Group and Firewall Setup

Ensure the instance's security group allows inbound access to ports 22, 80, and 443 for both IPv4 and IPv6. Also ensure ICMP is allowed to enable ping.

7. Managing Services (for future use)

To manage either service (frontend or backend):

Start: sudo systemctl start [service-name]
Stop: sudo systemctl stop [service-name]
Restart: sudo systemctl restart [service-name]
Status: sudo systemctl status [service-name]

Replace [service-name] with either 'frontend' or 'ecgscan-backend'.

8. Accessing the Application

Once everything is set up and both services are running, access the system using the server's IPv6 address in the browser.