



AI-VVO sdmay22-36  
Spring 2022  
Update #7

---

3/10/2022-3/24/2022



# Front-end (This Week)

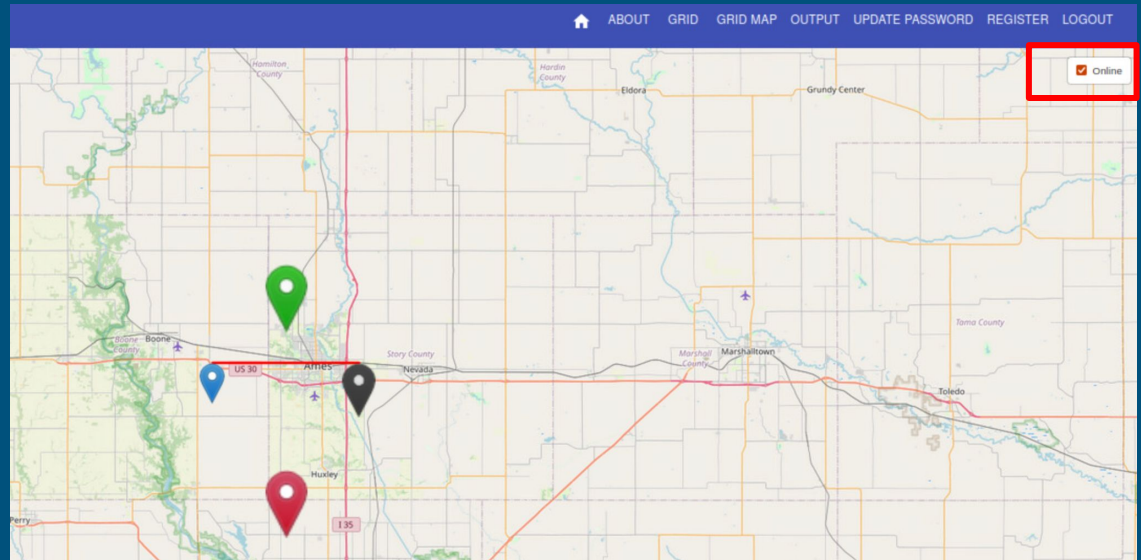
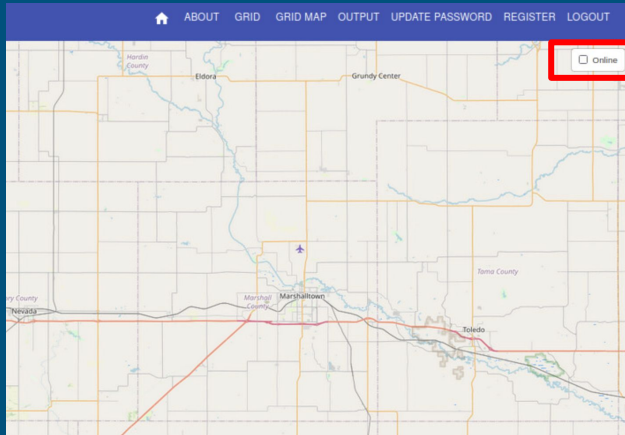
- Added a “Layer Control” to the map
- This will allow toggling of different busses/nodes

```
return(  
  <MapContainer className="map"  
    center={position}  
    zoom={10}  
    maxZoom={30}  
    style={{ height: 750, width: "100%" }}  
  >  
    <LayersControl position="topright">  
      <LayersControl.Overlay checked name="Online">  
        <LayerGroup>  
          {locations.map((location) => (  
            //if (location.iType == 'Online') {  
  
            <Marker  
              position={location.position}  
              //icon={defIcon}  
              icon={GetIcon(location.size, location.iType)}  
              onClick= { () => {  
                const token = localStorage.getItem('token');  
                axios.post(`${settings.API_SERVER}/api/auth/marker/`, {  
                  data: 'Hello World'  
                }, {  
                  headers: {'Authorization': `Token ${token}`}  
                })  
              }  
            )  
          )  
        }  
      </LayersControl.Overlay>  
    </LayersControl>  
  </MapContainer>  
)
```

# Front-end (This Week)

- Will need to be updated to filter based on node/bus type

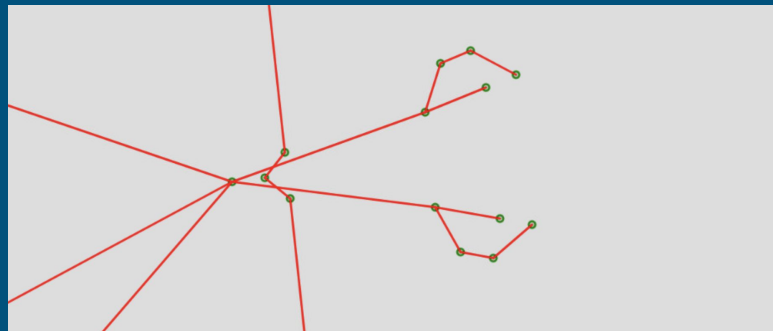
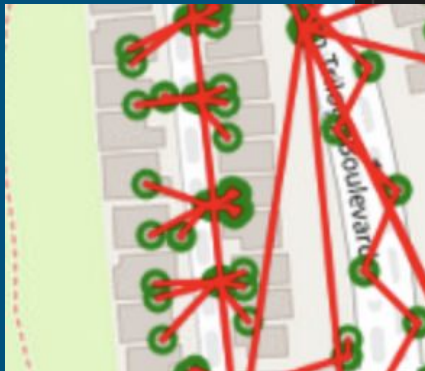
REGISTER LOGOUT



# Front-end (This Week)

- Increased max zoom on OpenStreetMaps
- Will need to be adjusted so actual map renders

```
return(  
  <MapContainer className="map"  
    center={position}  
    zoom={17}  
    maxZoom={30}  
    style={{ height: 750, width: "100%" }}  
    preferCanvas={true}
```



# Front-end (Next Week)

---

- Continue implementing updates to user interface
- Change lines to dashed when switch is flipped (ajax)
- Implement Solar panel icon
- Figure out how to distinguish between PV nodes and other nodes

# Back-end (This Week)

---

- Made improvements into the update\_switch endpoint. Now calls the scripts made by the Machine Learning team
- Validate inputs into the update\_switch endpoint to ensure bad data isn't given to the ML scripts

# Back-end (Next Week)

---

- Look into ways to improve database query speeds to help make frontend more responsive
- Look into possibly introducing batching into the neo4j queries to make pulling data less intense for the frontend

# Machine Learning (This Week)

---

- Protection.dss file can now be updated with the toggled switch state using the SwitchUpdate.py and ParseLineData.py scripts
- All components fully integrated now
- Machine learning information can be used to interact with the backend database and modify the switch states of elements in the OpenDSS instance



# Machine Learning (Next Week)

---

- Refine control signal in the ML algorithm to choose correct element to toggle
- Work on main loop that will take control signal and will determine future actions
- Work on adding functionality to control the regulators and capacitor banks
- Currently only switches can be modified