



# AI-VVO sdmay22-36 Weekly Update #2

---

10/4/2021 - 10/10/2021



# Front-End

---

- Completed more in-depth ReactJS tutorial at [https://www.youtube.com/watch?v=Ke90Tje7VS0&t=1830s&ab\\_channel=ProgrammingwithMosh](https://www.youtube.com/watch?v=Ke90Tje7VS0&t=1830s&ab_channel=ProgrammingwithMosh)
- Learned about how components and properties work and understand better how to structure overall application
- Cloned previous team's repository to local machines and reviewed the code and attempted to get it running.
- Added specification to Frontend Requirements

# Front-End

---

## Requirements Updates:

- Algorithm selection on the UI screen
  - Add dropdown menu selection to pick which algorithm to run
  - Add algorithm selection to output page and report
- Map-based power grid display
  - Implement Google Maps on a separate page
  - Add nodes overlay onto map
- Update UI to be more user friendly
  - Change the styles of the buttons and sliders
  - Make on/off slider a switch instead of a button
  - Add information on how to use the button and sliders so new users can understand what they are changing
- Update output page to display data in a more readable format
  - Add option to download report
  - Make report more readable using a grid display
  - Make the slider/button side of the screen smaller
  - Make simulate a button so they can rerun test
  - Add the regulator value to the regulator position so the user knows what they tested
- Add voltage information of nodes to grid display page
  - Add voltage data to properties of node and only visualize in a popup when user clicks on node

# Front-End (Plan for Next Week)

---

- Create a high level diagram / overview of the structure of the frontend application using class components
- Continue review of the previous group's code and get running on local machines
- Write up documentation on how to run the app on a local machine
- Continue ReactJS tutorials for specific components of our application including D3-Graph database for data visualization

# Back-End

---

- Researched and did tutorials on Django and Python
- Started to look into PostgreSQL
- Continued to look into previous teams code and the resources they left for us
- Basic research into machine learning to try to understand the machine learning api
- Looked into ways to improve documentation/readability in previous teams code

# Back-End (Plan for next week)

---

- Continue to do Django and python tutorials
- Research and practice with PostgreSQL
- Try to find solutions to improve Django database connection
- Try to get the previous teams code working locally
- Increase the documentation on the backend to prepare for any future changes

# Machine Learning (Research)

---

- Analyzed sdmay21-24 code further
- Familiarized ourselves and dived deeper into PyTorch documentation and deep convolutional neural network tutorials
- Researched implementing deep convolutional neural networks in PyTorch
- Watched a great video on how to use PyTorch's built-in tools to create layers in the neural network. This video is included below.
- torch.nn module allows for creating convolution layers, pooling layers, fully connected layers, and putting these layers through nonlinear activation functions
- May need to investigate normalizing the input data from each bus. May have an exploding gradient problem if the input values aren't between 0 and 1.

[https://www.youtube.com/watch?v=pDdP0TFzsoQ&ab\\_channel=PythonEngineer](https://www.youtube.com/watch?v=pDdP0TFzsoQ&ab_channel=PythonEngineer)

# Machine Learning (Findings)

---

- Convolutional networks to approximate the Q function will be a good approach for controlling the power distribution system.
- Want input vector of model to be the inputs from the CSV files
- Want to learn the correct position of the three voltage regulators and the cap bank. This will be the weight matrix  $w$ .
- Output will be new real and reactive power values for all feeders as a consequence of changing the control mechanism positions
- Next week, we will dive into control documentation to start to implement Q learning with convolutional network