Things are listed in order or priority. (First item is most urgent, last is least urgent). These are the most urgent tasks at hand, if there are more job in the future, we will pay more accordingly.

**IMPORTANT: Must be familiar with MERN stack, integration of data from websockets and api, authentication**

**Authentication**

1. Signup, login, reset password
   1. Done with signup and login only, need help showing error messages when invalid credentials are keyed in
2. Routing based on email credentials (Roles)
   1. Done with routing based on roles extracted from email (example: [person@dbs.sg](mailto:person@dbs.sg) has a role of ‘dbs’ as I just take the string after ‘@’ and before ‘.’)
3. For authentication, everything seems to be working well until just now when I tried to deploy (couldn’t sign in) and I have problems with debugging, may need your help on that.

**Charts (2 urgent charts for now)**

1. Fetch real live data form websockets / api and push them into existing frontend charts.
   1. Done with pushing mock data (json format)
2. Enable users to view chart from previous day / month etc.

**Live update of numbers / status (around 5 components)**

1. Other than charts, we have text fields where numbers should be live as well.
   1. Done with the positioning of numbers. Need help with updating the numbers at intervals.
   2. We also want to have arrows that change colour or direction based on the live data.
   3. We have different colours correspond to different status
      1. If data from the websocket says that the IoT is ‘active’, it should show a green circle on frontend.
         1. Hardcoded the colours for now.

**Exporting data button**

1. A button to export / download the current data we have in the database

**Frontend**

1. Clean up
   1. I have hard coded the components according to pixels to fit 16:9 screen, but looks dis-proportionate in other window sizes. Best if it can adjust to different window sizes. However this is not of high priority because the display we will use for the website is a 16:9 tv, in fullscreen (which is what I referred to when hard coding the positions)