# ICE Reader

*Internal Combustion Engine Reader*

Fredrick Collins

*v0.1*

**Project Description**

I am proposing a plug-and-play solution to fault code reading, real-time feedback, data logging, and analysis for the commercial vehicle via the OBD-II port through ICE (*Internal Combustion Engine)* Reader. Utilizing a Raspberry Pi Zero for computing, ICE Reader will operate by drawing power from any 12V plug and interfacing with the ECU (engine control unit) over micro-USB to query and return a range of useful values. After scanning for standard fault codes on startup and recording the response, ICE Reader will power up an on-board OLED display complete with a joystick and two buttons to load a start menu. From the menu, users can start/stop data logging or select a real-time readout of values such as speed, intake pressure, rpm, and more.

This solution doesn't stop when the car does, because if the user wishes to view and manipulate a variety of data series they have logged (such as those mentioned above), they can remove the micro SD card from the Raspberry Pi, insert it into their computer, and *visit the ICE Reader website* to chart, compare, *request reports*, and save/load recorded data from our database as they please.

*Note: This project will be expanding upon work I had done in my senior year of high school. Previously, I had established a connection between the Raspberry Pi and the ECU, queried intake pressure, and displayed it to the screen in real time - the buck stopped there. All user inputs via menu, queried values, data logging, data exporting, and charting features will be new.*

**Features**

- Read and record common ECU fault codes
- Start and stop recording a wide array of engine variables in real time via user interface
    - Display a wide array of engine variables in real time via OLED screen
- Ability to create an account and log in to gain access to your saved files
    - Ability to reset password or delete account
- Charting capability supporting multiple series on the same x-axis
    - Adjustable display scale, automatic high/low labeling
    - Ability to add moving averages to smooth data
- Export capability to save entire or certain parts of data logs to ICE database

**Users**

- Car enthusiasts
    - Measure timing advance, turbocharger pressure, etc.
- Regular consumers
    - Check for trouble codes, measure temperature, etc.
- Insurers or parents
    - Monitor speed, throttle usage for safety concerns, etc.

**Technology**

The Raspberry Pi Zero will be running Raspbian, a linux operating system for Raspberry Pi’s. The ICE Reader software will be written in Python 3, and utilize a library for I2C connection and OBD-II protocol. Ext2explore will be used to unpack data from the micro SD onto a windows computer, *and PHP with chart.js will be used to interact with a MySQL database and provide a user interface and charting application on a website hosted by Heroku.*

**Scope**

**<u>/home/pi/ice/</u>**

**-menu.py**

*contains code to render and make functional main menu, where user can utilize joystick and two buttons on the hardware unit to select from any four of the programs below*

**-diagnostics.py**

*simplest program in the menu, press button to select, and a loading screen flashes quickly while the reader runs an error code query and displays the response. press any button to go back to menu*

**-display.py**

*after running, a submenu with choices like rpm, speed, intake pressure, temp, etc. is shown. any of these can be selected and from there the screen will clear to show a live readout of that value. press any button to go back*

**-log.py**

*after running, displays just the status of the program. will write the full array of recorded values to the sd card at intervals. press any button to go back*

**-web.py**

*runs log.py, but pushes data to locally hosted web server at intervals. user can view on their device by connecting to the pi's address*

**-export.py**

*searches for internet connection, pushes log data to mysql server. deletes after a successful response*

Heroku will be used to host the web pages, and the setup should look something like this:

**<u>heroku/ice/</u>**

**-explorer.php**

*the data exploration homepage. mocked up in prototype*

**-login.php**

*handles login authentication, doubles as landing page. has redirection to signup / reset password*

**-signup.php**

*handles new user signup, password verification.*

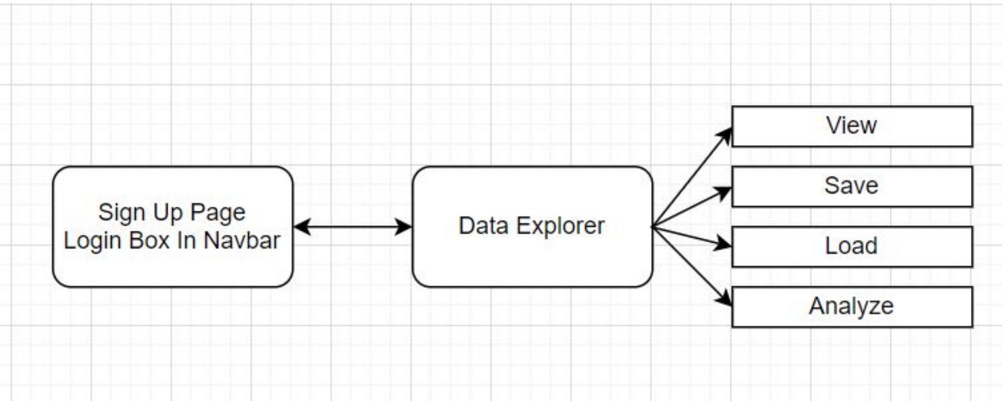
**-upload.php**

*handles .csv uploads, pushes to mysql server*

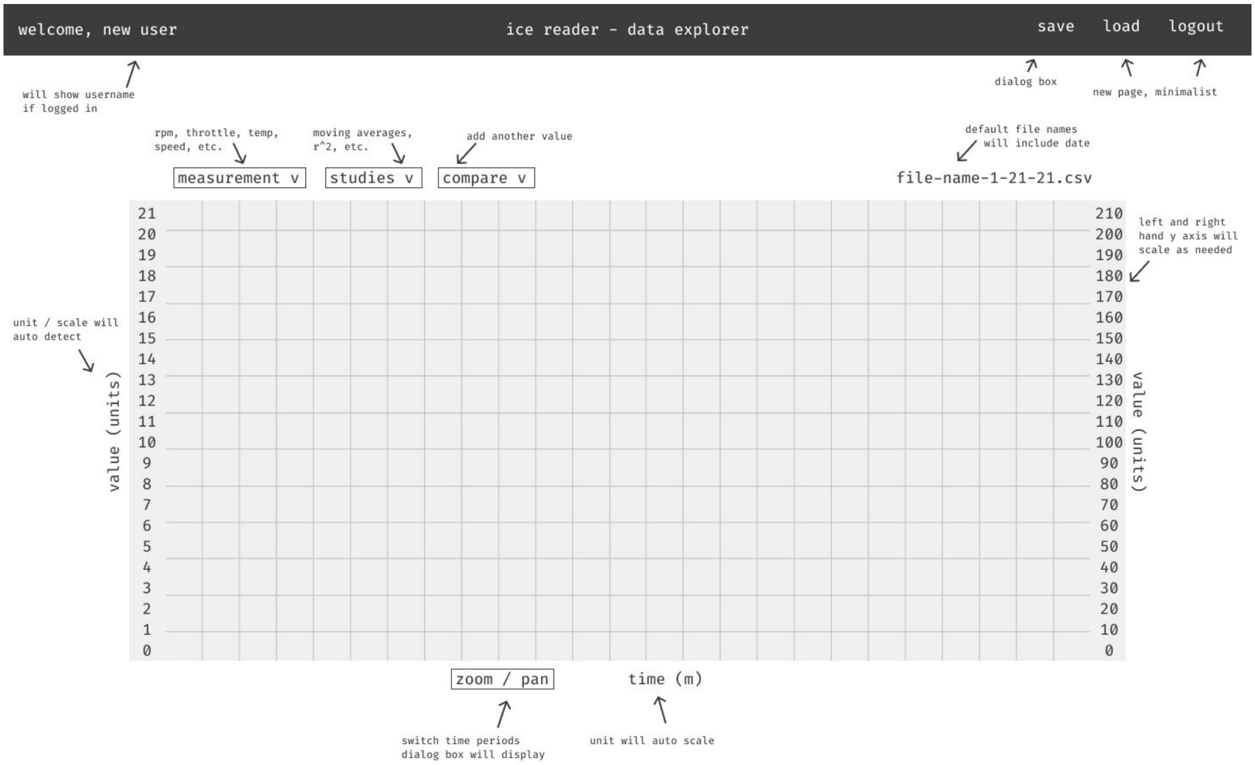
**-load.php**

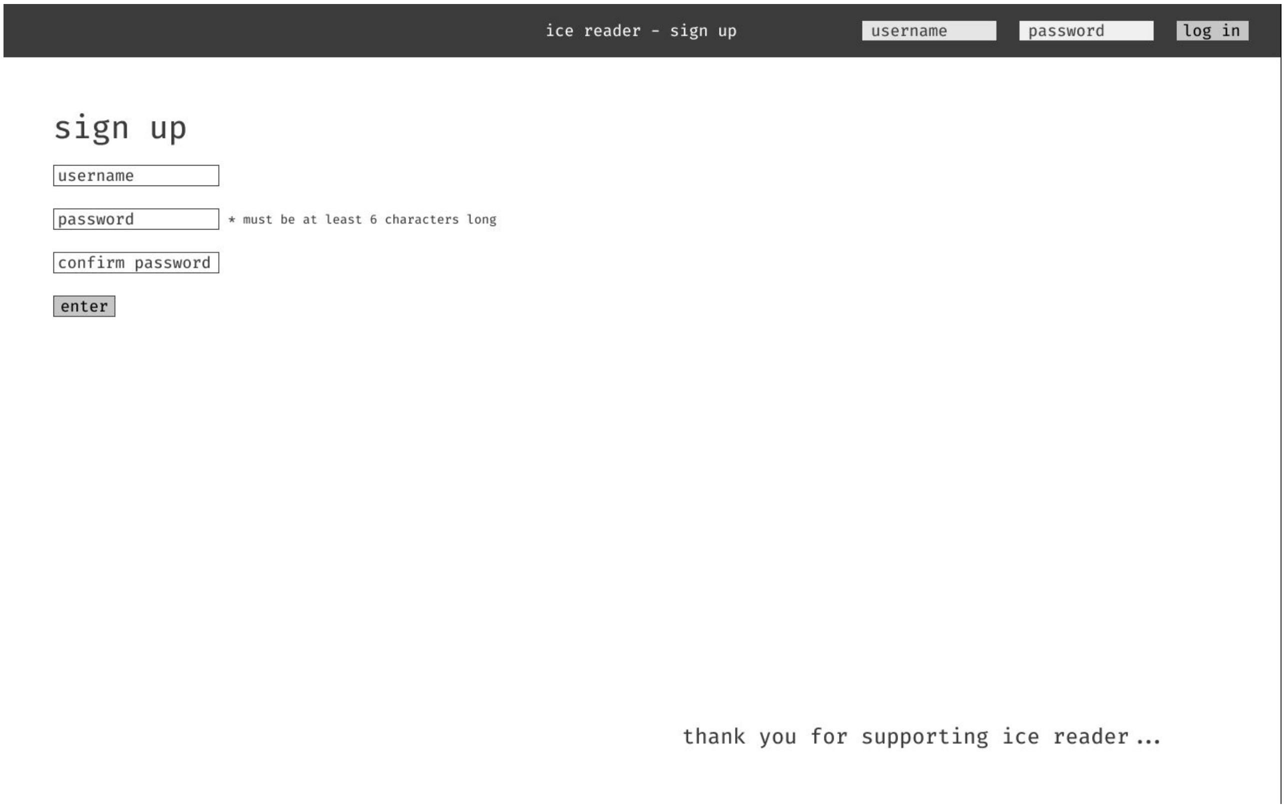
*pulls from mysql server to load into data explorer*

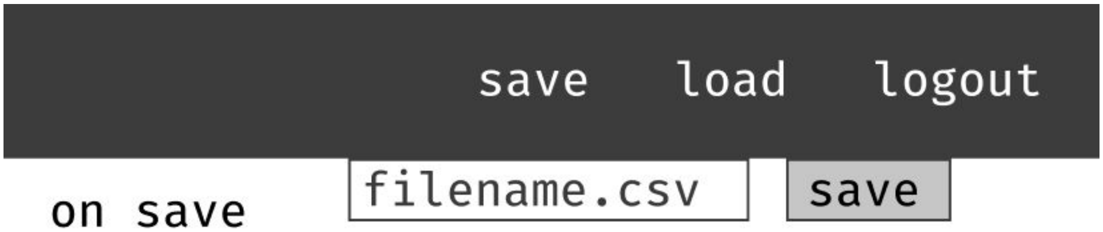
### Site Flowchart



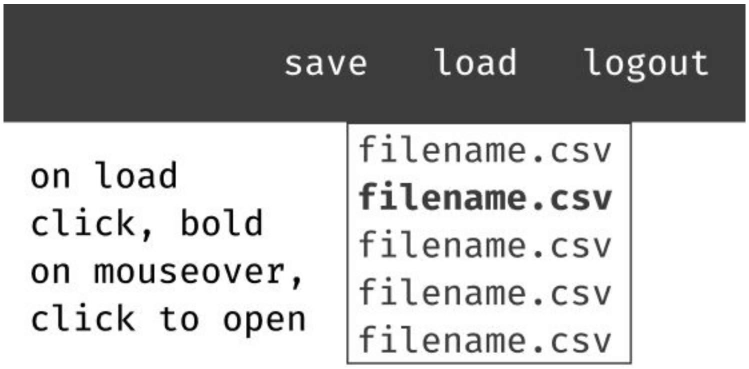
# Wireframes


welcome, new user
ice reader - data explorer
save
load
logout
will show username if logged in
dialog box
new page, minimalist
rpm, throttle, temp, speed, etc.
moving averages, r^2, etc.
add another value
default file names will include date
measurement v
studies v
compare v
file-name-1-21-21.csv
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
left and right hand y axis will scale as needed
unit / scale will auto detect
value (units)
value (units)
zoom / pan
time (m)
switch time periods dialog box will display
unit will auto scale


ice reader - sign up
username
password
log in
sign up
username
password
* must be at least 6 characters long
confirm password
enter
thank you for supporting ice reader...



on save click



on load click, bold on mouseover, click to open