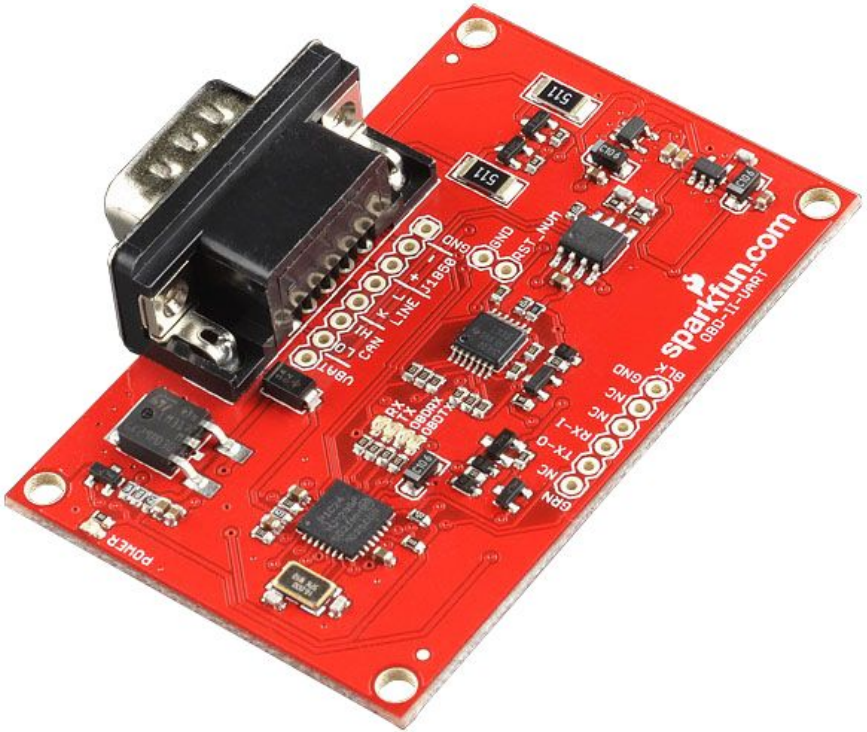
CSCI 3308: smartOBD.

Team 2: CodeHawks

Travis Cochran
Fallyn Logan
Avra Saslow
Brett Schneider
Will Walker
Andrew Weller

OBD-II


MS309
Diagnostic Menu 1/6
1)Read Codes
2)Erase Codes
3)View Freeze Frame
4)I/M Readiness
OBD
OKGORD
CAN OBD II SCAN TOOL

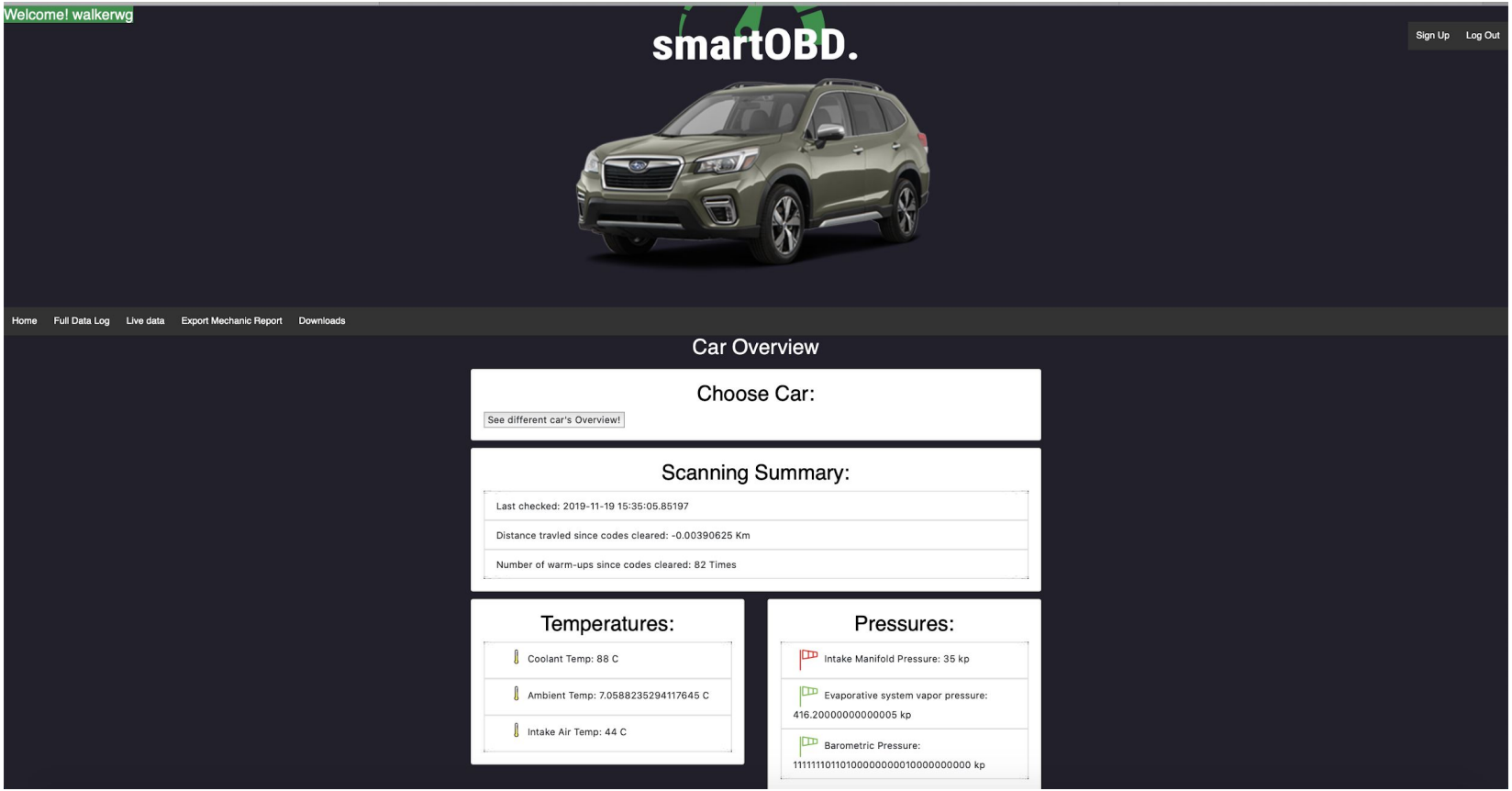


- Cheap
- Customizable
- Ease of access and use

Vision Statement

For automotive tinkers who need a flexible OBD-II scanner for unique use cases. The smartOBD is an OBD-II scanner that is customizable and open source.

Features


Welcome! walkerwg
smartOBD.
Sign Up
Log Out
Home
Full Data Log
Live data
Export Mechanic Report
Downloads
Car Overview
Choose Car:
See different car's Overview!
Scanning Summary:
Last checked: 2019-11-19 15:35:05.85197
Distance traveled since codes cleared: -0.00390625 Km
Number of warm-ups since codes cleared: 82 Times
Temperatures:
Coolant Temp: 88 C
Ambient Temp: 7.0588235294117645 C
Intake Air Temp: 44 C
Pressures:
Intake Manifold Pressure: 35 kp
Evaporative system vapor pressure: 416.20000000000005 kp
Barometric Pressure: 11111110111010000000010000000000 kp

Process

1. Python script of OBD commands: ~200
2. Filling database with car information
3. Create template pug/jade website
4. Add dropdowns/buttons
5. Async, logs, home (NodeJS)
6. Styling pages and images
7. Adding JavaScript Widgets

04	ENGINE_LOAD	Calculated Engine Load	Unit.percent
05	COOLANT_TEMP	Engine Coolant Temperature	Unit.celsius
06	SHORT_FUEL_TRIM_1	Short Term Fuel Trim - Bank 1	Unit.percent
07	LONG_FUEL_TRIM_1	Long Term Fuel Trim - Bank 1	Unit.percent
08	SHORT_FUEL_TRIM_2	Short Term Fuel Trim - Bank 2	Unit.percent
09	LONG_FUEL_TRIM_2	Long Term Fuel Trim - Bank 2	Unit.percent
0A	FUEL_PRESSURE	Fuel Pressure	Unit.kilopascal
0B	INTAKE_PRESSURE	Intake Manifold Pressure	Unit.kilopascal
0C	RPM	Engine RPM	Unit.rpm
0D	SPEED	Vehicle Speed	Unit.kph
0E	TIMING_ADVANCE	Timing Advance	Unit.degree
0F	INTAKE_TEMP	Intake Air Temp	Unit.celsius
10	MAF	Air Flow Rate (MAF)	Unit.grams_per_second
11	THROTTLE_POS	Throttle Position	Unit.percent
12	AIR_STATUS	Secondary Air Status	string
13	O2_SENSORS	O2 Sensors Present	special
14	O2_B1S1	O2: Bank 1 - Sensor 1 Voltage	Unit.volt
15	O2_B1S2	O2: Bank 1 - Sensor 2 Voltage	Unit.volt

Tools Overview

Methodology: **Waterfall** ★★★

Communication Line: **Slack** ★★

VCS Repository: **Github** ★★

Hardware: **Sparkfun OBD-II Board** ★★★★★

Framework: **Node.js** ★★★★

Database: **PostgreSQL** ★★★★

Testing Tool: **Pytest, TravisCI** ★★★★

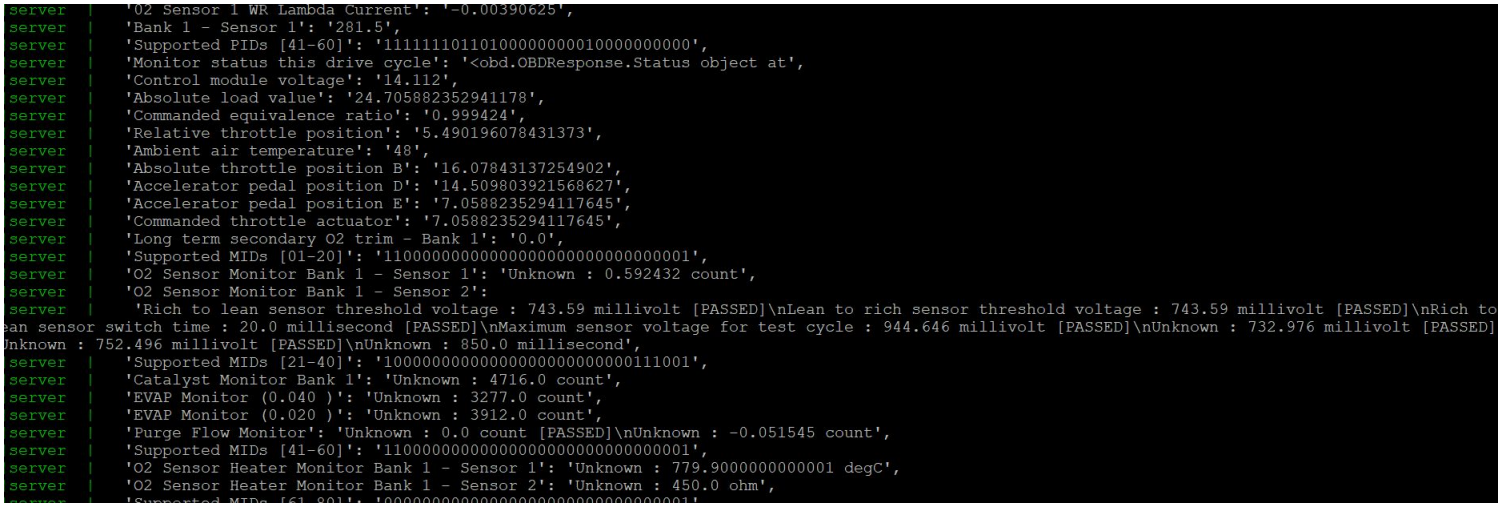
Deployment: **Ubuntu** ★★★





Challenge 1: Parsing Through Log Values

- Different car brands support different OBD-II features
  - Dealing with the variety of returned commands, and the number of available commands in the client-side software was difficult



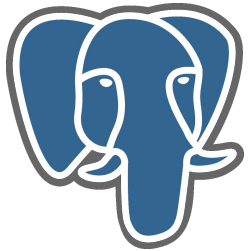
Unit.percent
Unit.celsius
Unit.percent
Unit.percent
Unit.percent
Unit.percent
Unit.kilopascal
Unit.kilopascal
Unit.rpm
Unit.kph
Unit.degree
Unit.celsius
Unit.grams_per_second
Unit.percent
string
special
Unit.volt
Unit.volt

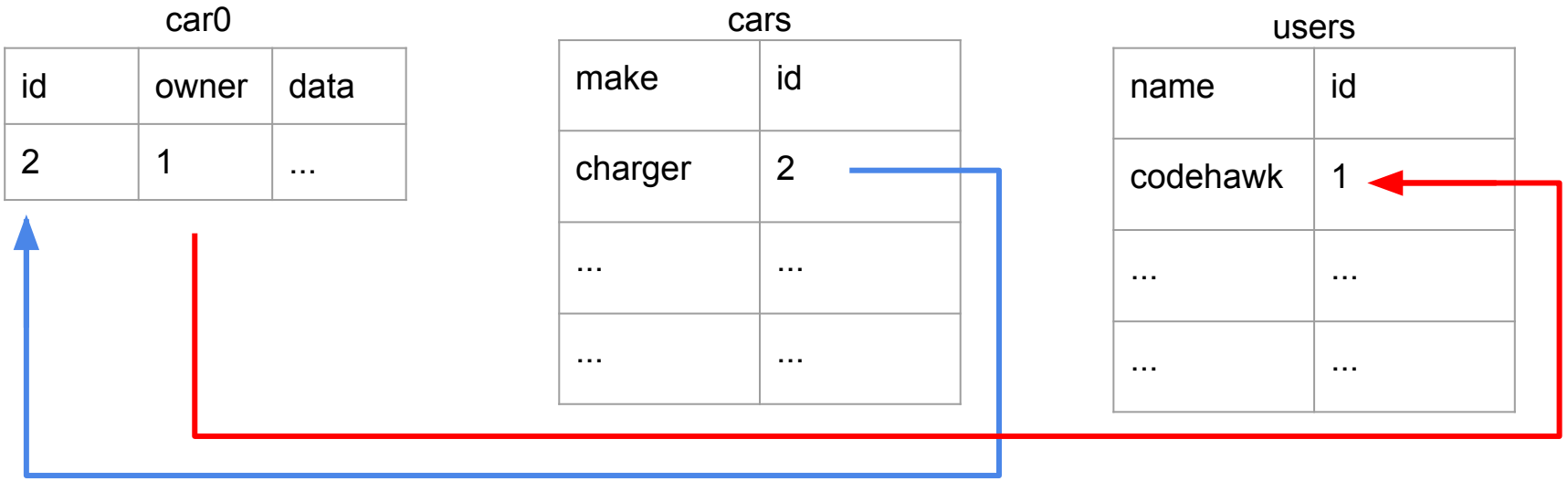
Challenge 2: Async and Full Log

- Writing nodeJS middle-ware for full-log, live-data, and login was quite challenging.
- Had to connect user accounts with data read in dynamically by client program.
- Took more project time than expected

Challenge 3: Database

- PostgreSQL
- Users Table - points to individual user tables
- Cars Table - points to individual car tables





Challenge 4: Javascript Widgets

- Used to visualize some of car data on home page
- Hard to interpret raw car data
- Lots of casting and converting
- Caused less overall widgets than planned

Final Product

http://198.23.146.166:3000/home

Skype Live Experience

Sponsored by Arby's