The following are the files used in the project and a description of their purpose:

pi.env	AWS credentials for Raspberry Pi.
pyserver.py	Flask server that runs on the Raspberry Pi. Listens for motion detection events and lock/unlock commands from the EC2 server.
main.py	Script that sets up the doorbell and listens for button presses.
lock.py	Script for locking and unlocking the door with the servo.
hodoor.service	Systemd service that handles the setup of the ssh tunnel
David.env	Environment file that sets up several command line variables related to an AWS IOT thing. Used in testing.
ec2/ec2.env	Environment file that sets up several command line variables related to an AWS IOT thing. The ec2 server was registered as an AWS IOT thing and this environment file sets up its parameters
ec2/elm-package.json	Elm module dependency list
ec2/package.json	Nodejs module dependency list
ec2/webpack.config.js	Script that takes all source files, and forms a minified bundle serving as a single webpage.
ec2/webpack.parts.js	Script used in conjugation with webpack.config.js to generate a *source map* file. Source maps are used by browsers to map lines of a minified file to source code. A bit like the .text section in an executable.
ec2/src/index.js	Entry point for webpack. Attaches the app to index.html.
ec2/src/api.js	Code for the node server run on the ec2.
ec2/src/index.html	HTML file that the node server serves to clients. Acts as the document root for the app.
ec2/src/*.elm	We used ELM as our website framework. Elm is a Haskell that compiles down to a single javascript file. This javascript file attaches itself to a dom node of index.html and serves the app.
ec2/src/LF/*.elm	Elm files specifically related to the login screen.