**New Orleans Release Date Calculator**

**The Legal Need:**

The criminal justice system is riddled with inefficiencies, disorganization, and outdated processes that threaten individual liberty and justice itself. Incarceration rates in the United States are at an all-time high and Louisiana is no exception to this statistic. In fact, Louisiana currently posts the highest incarceration rates in the country with 35,000 people currently incarcerated. Limited legal resources and systematic over-detention plague the Louisiana justice system. Public defenders are overworked and under-resourced. They are forced to counsel clients with little time to prepare and currently don’t have the necessary tools to accurately and efficiently advise clients regarding their release eligibility dates during sentencing or plea negotiations. Furthermore, these overworked and understaffed defenders’ offices are unable to adequately track their clients post incarceration to ensure clients are considered for parole or released under good time or regular release when they are eligible. Outdated logging systems further exacerbate this issue leading to system-wide over-detention.

This application is designed to help determine when an individual who is or will be incarcerated will be eligible for good time release, when they will be eligible for parole, and what their last possible day of release is or will be. The application is designed with two different types of users in mind: public defenders and family members of incarcerated individuals. These two kinds of users are using the application in different ways. A public defender is primarily using this application to advise their clients during sentencing or plea deal negotiations. A family member is primarily using this application once their loved one is already serving time to plan for their loved one’s release. Enabling public defenders to use this application will aid them in navigating Louisiana’s complicated and often counter-intuitive sentencing laws so that they can give clients a full and accurate picture of their options and potential outcomes prior to sentencing. Enabling family members to use this application will serve two purposes. The first is in helping address the issue of over-detention of prisoners. Family members are likely the best resource to act as advocates for overdetained prisoners. This application will empower them to do so by giving them the information and resources they need. It provides family members with a way to determine when their loved one should be released with the hope that they ask questions and make noise when their loved one is held past that date. Secondly, providing family members with the ability to use this application will allow them to plan and prepare for their loved one’s release.

**An Institutional Purpose:**

This application is modeled off the New Orleans Release Date Calculator my team built for the Georgetown Law practicum entitled “Technology, Innovation, and Access to Justice.” In that practicum course, students, often with little to no programming experience, design and build applications to address pressing legal needs within the justice system on a platform called Neota Logic. That platform, while quite powerful as a tool to enable non-programmers to build legal applications, often requires lengthy logic constructions to implement functionality that could be programmed in a few simple lines in python. This project seeks to replicate the basic functionality of the New Orleans Release Date Calculation application built in Neota Logic as a way of evaluating the viability of adding a “programmers/python” track to the “Technology, Innovation, and Access to Justice Practicum.”

**Setup Guidance: (Also included in README.md)**

**Machine configuration:**

* If you don't have Homebrew, install it- terminal (anywhere)

/usr/bin/ruby -e "$(curl –fsSL

https://raw.githubusercontent.com/Homebrew/install/master/install)"

* Install Node/npm - terminal (anywhere)

brew install node

* if Github is not configured:

first step: <https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/>

second step: <https://help.github.com/articles/connecting-to-github-with-ssh/>

* Clone repository

git clone https://github.com/summerdanz/Nola-calc.git

* Install dependencies-terminal (inside project root)

cd Nola-calc

npm install

* Install dependencies- terminal (inside fullstack\_template/static)

npm install

* Install python3 - terminal (anywhere)

brew install python

* Use pip to install pipenv

pip3 install --user pipenv

sudo pip3 install virtualenv

* intall dependencies (cd fullstack\_template/server)

pipenv install requests

* activate virtual environment (inside fullstack\_template/server)

pipenv shell

* install flask (inside fullstack\_template/server)

pip3 install flask

* install flask\_cors (inside fullstack\_template/server)

pip3 install flask\_cors

**To run local development environment:**

* inside fullstack\_template/static:

npm run watch (ctrl + c to stop)

* In a new terminal window, inside fullstack\_template/server:

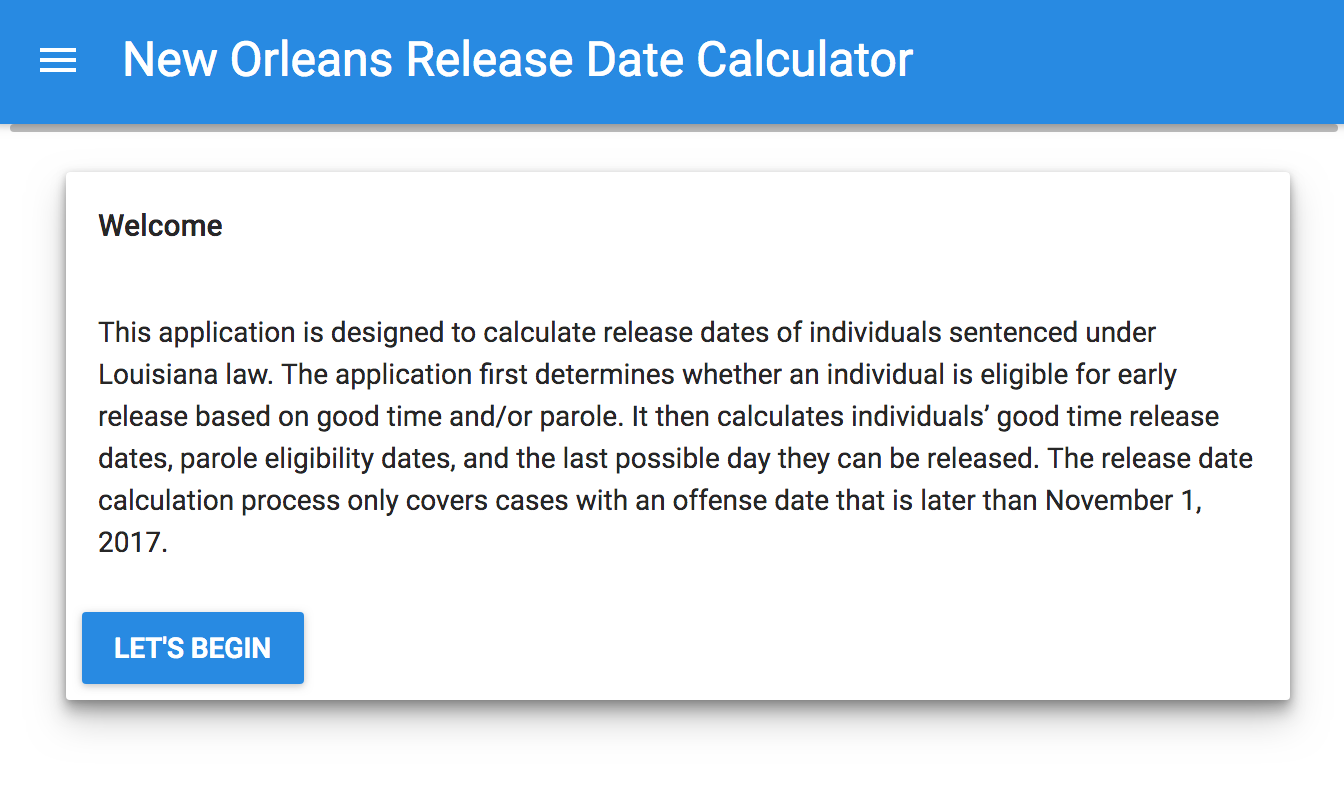
python3 server.py

* find project at localhost:5000

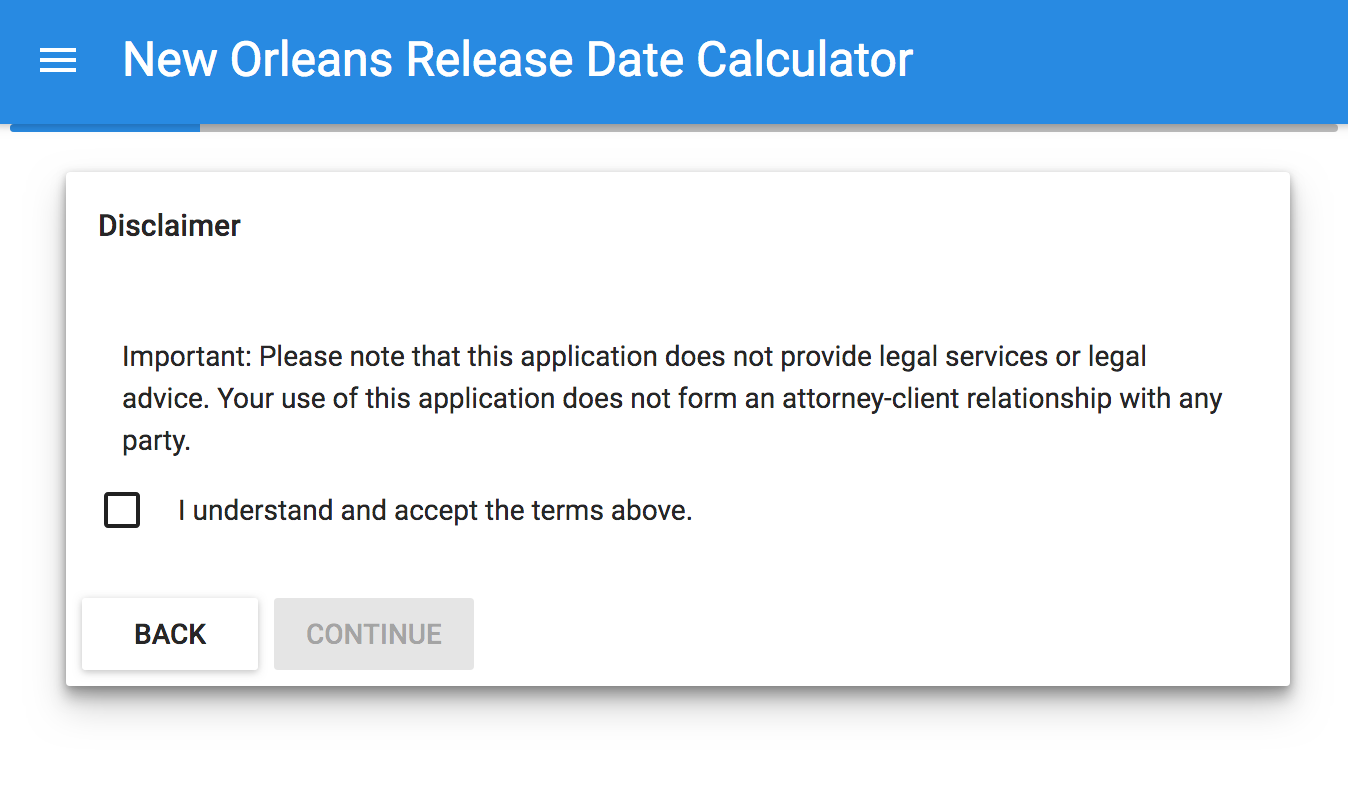
\*\* make sure cookies are disabled on this url in Google Chrome

**Application Overview:**

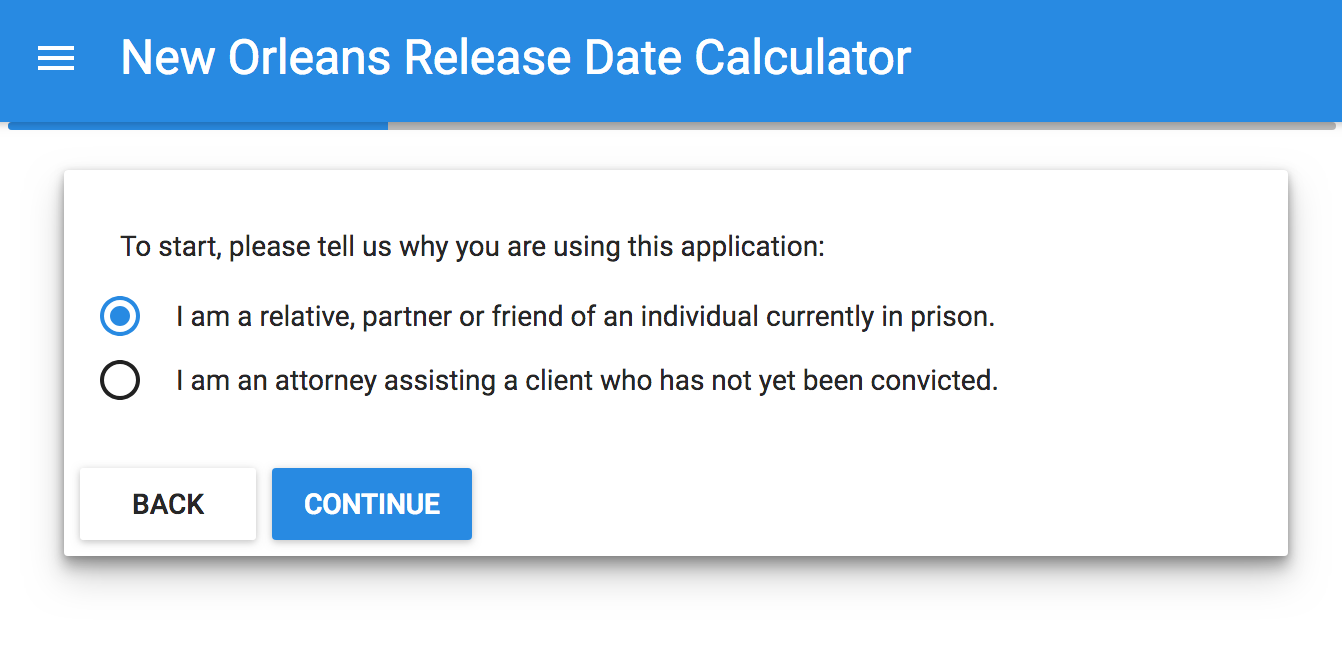
1. Introduction

****

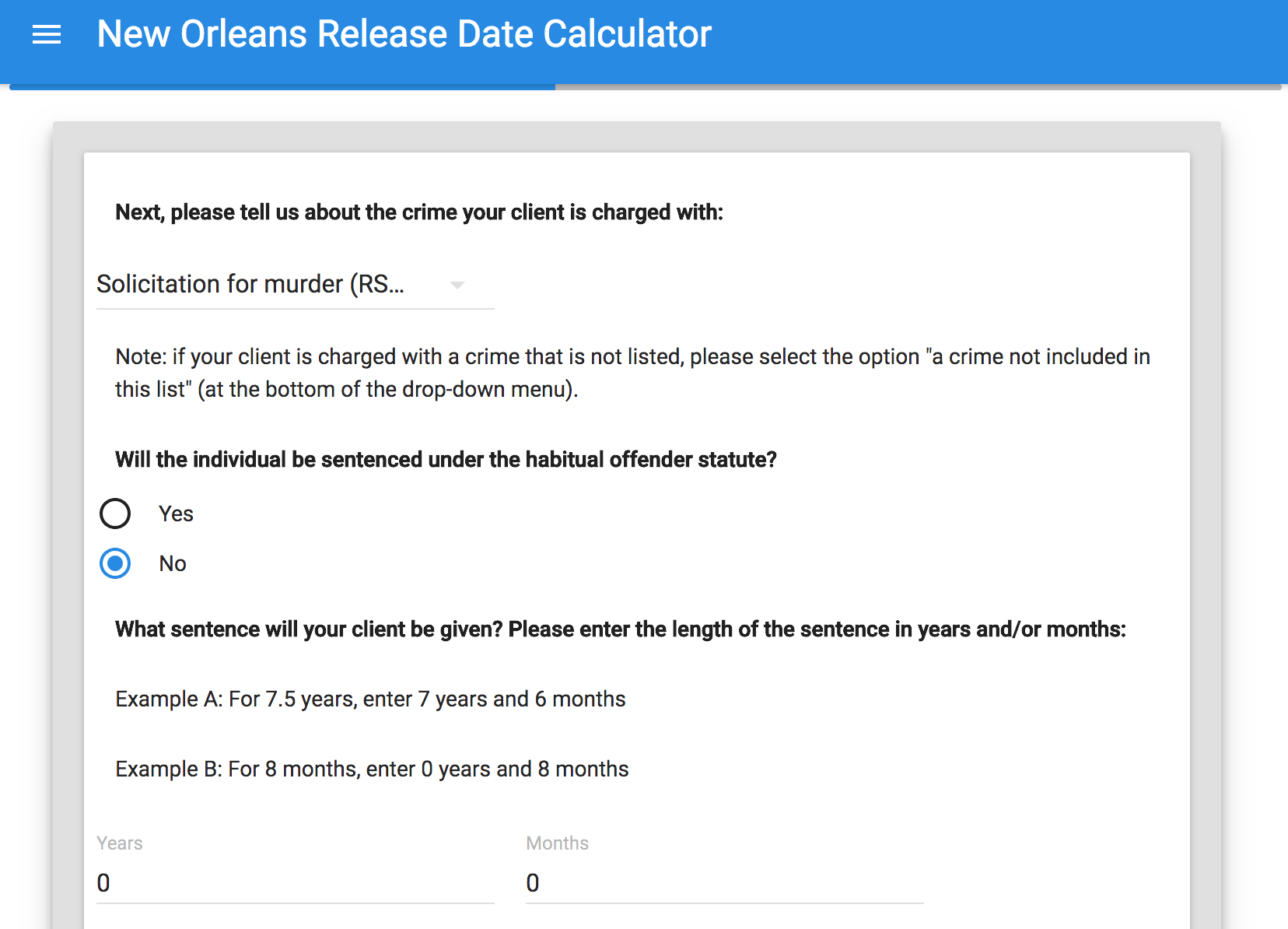
2. Disclaimer



3. Type of User

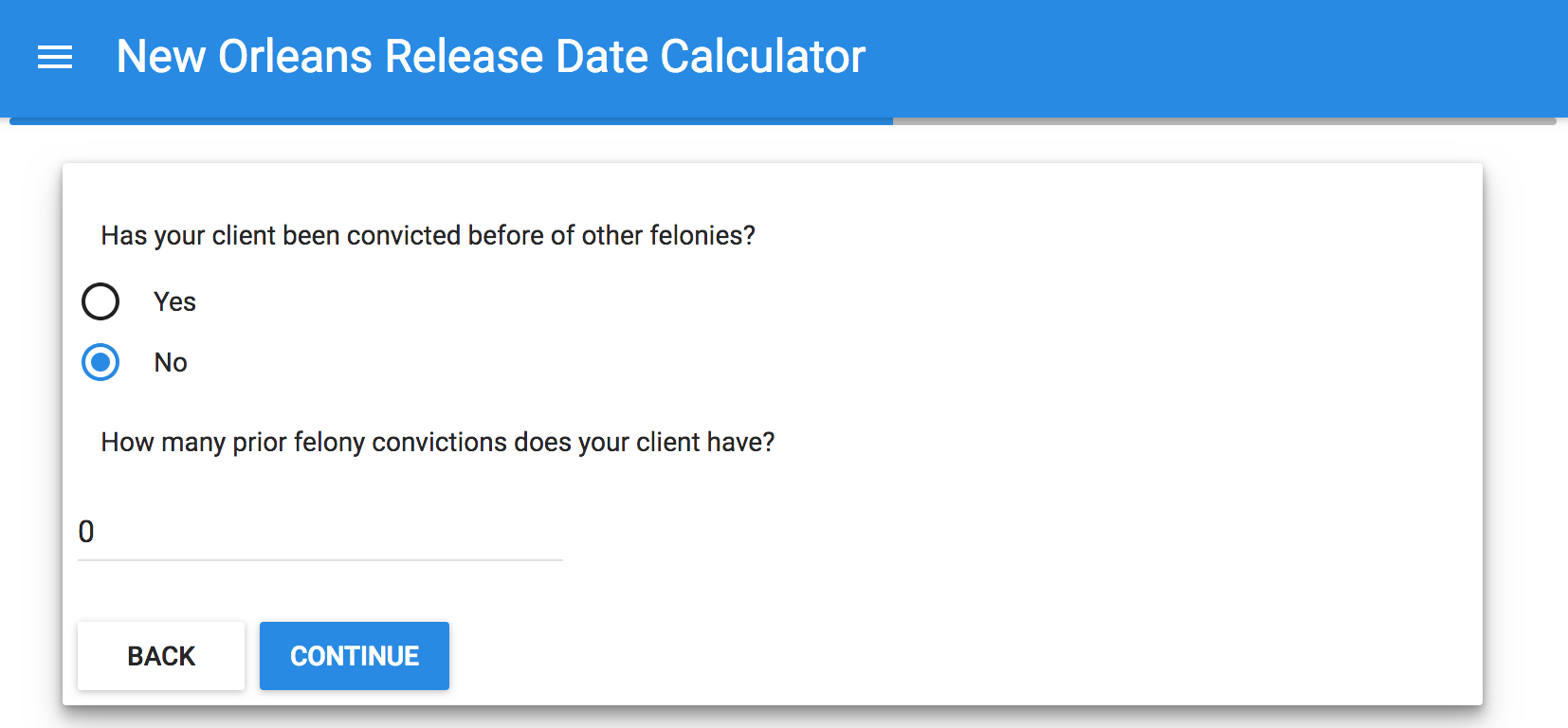
****

4. Current Charge Selection and Sentence Entry

****

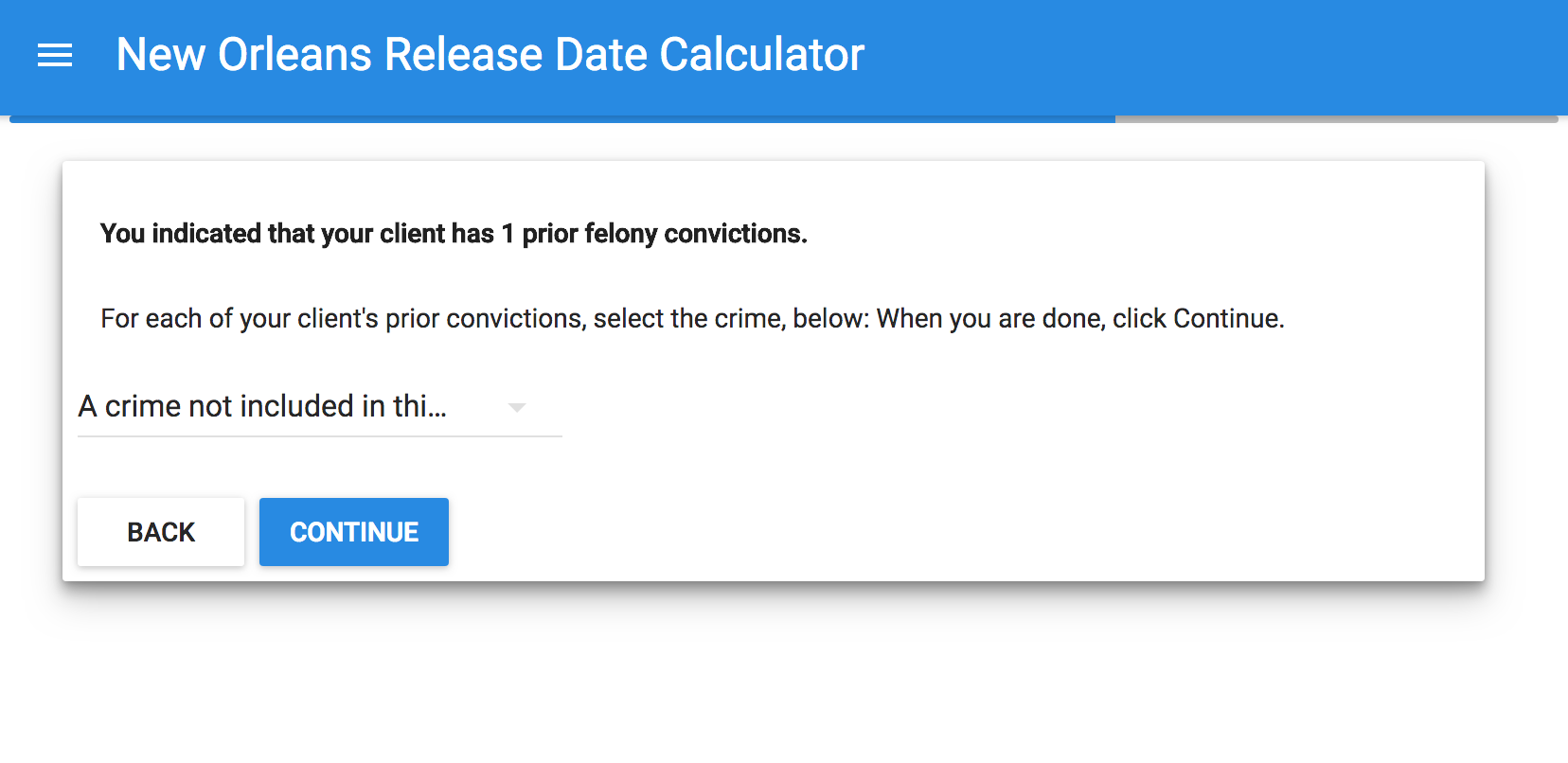
5. Has the individual been convicted before of other felonies?

The application asks the user if the individual/client has prior felony convictions. Prior felony convictions affect eligibility and timing of parole and good time release.

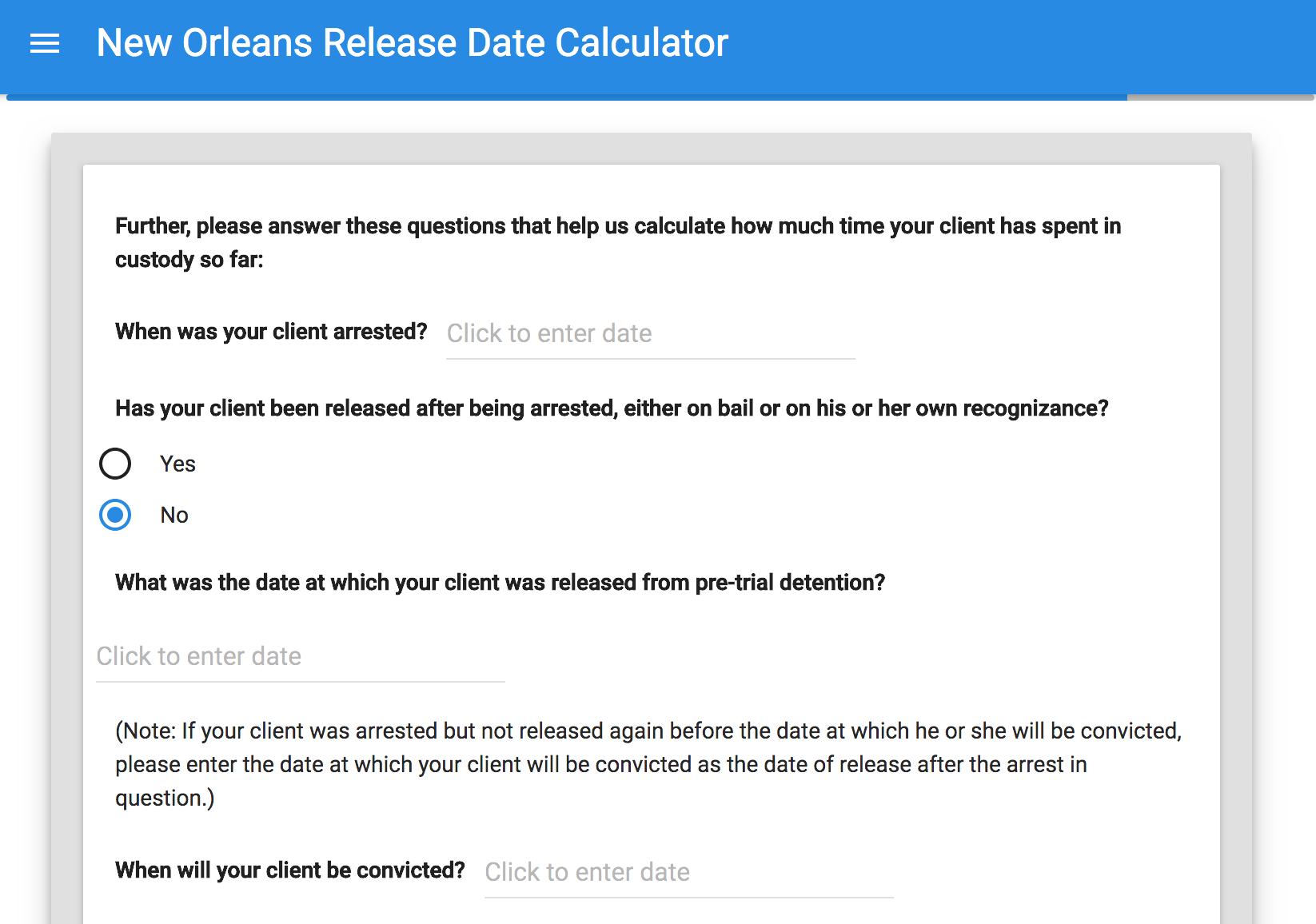


6. Prior convictions entry

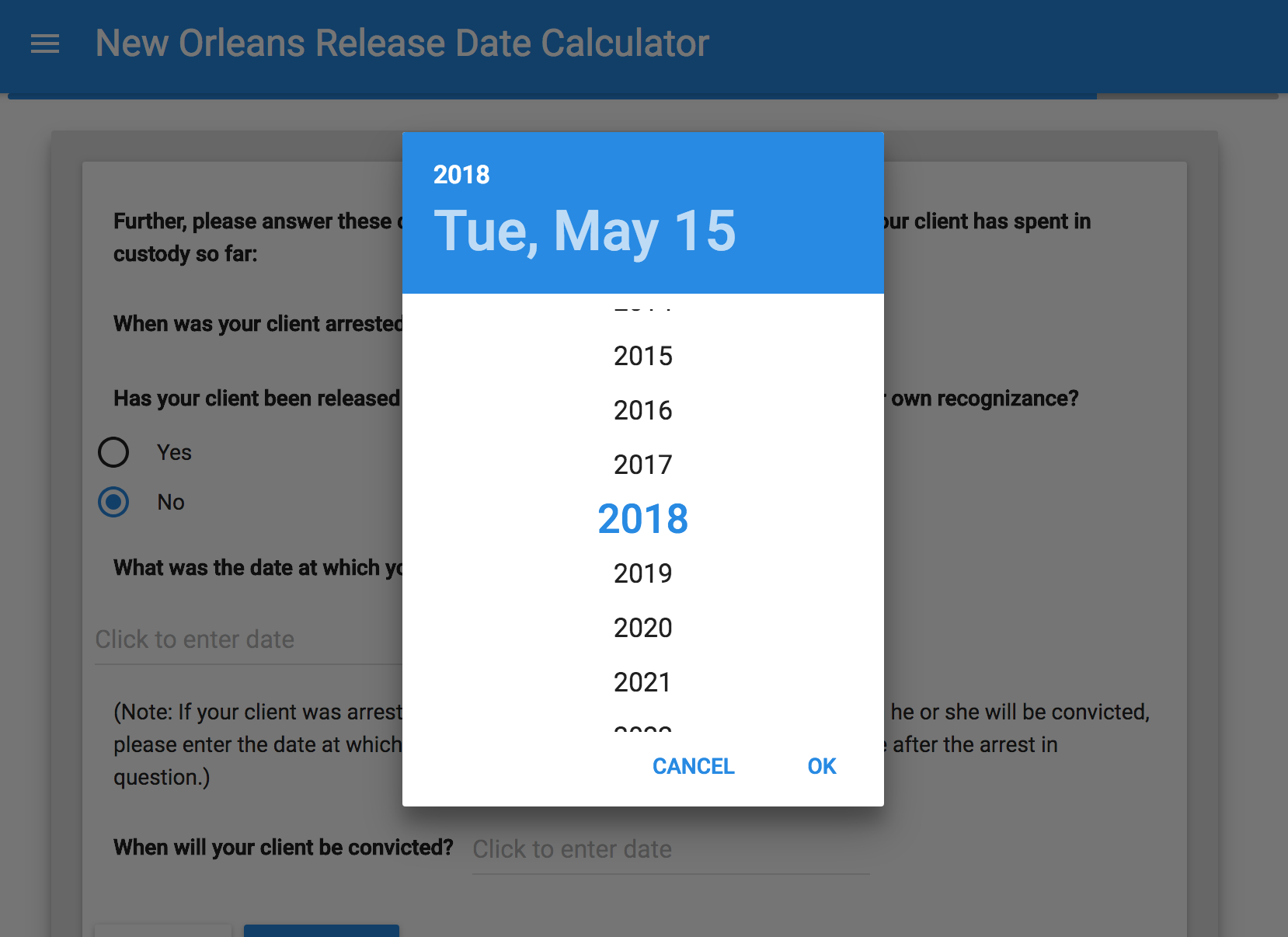
Users are prompted to select the individual’s prior charges from a drop down menu. The proper number of drop down menus populate based on the user’s answer in step 5 to “How many prior felony convictions does your client have?”

****

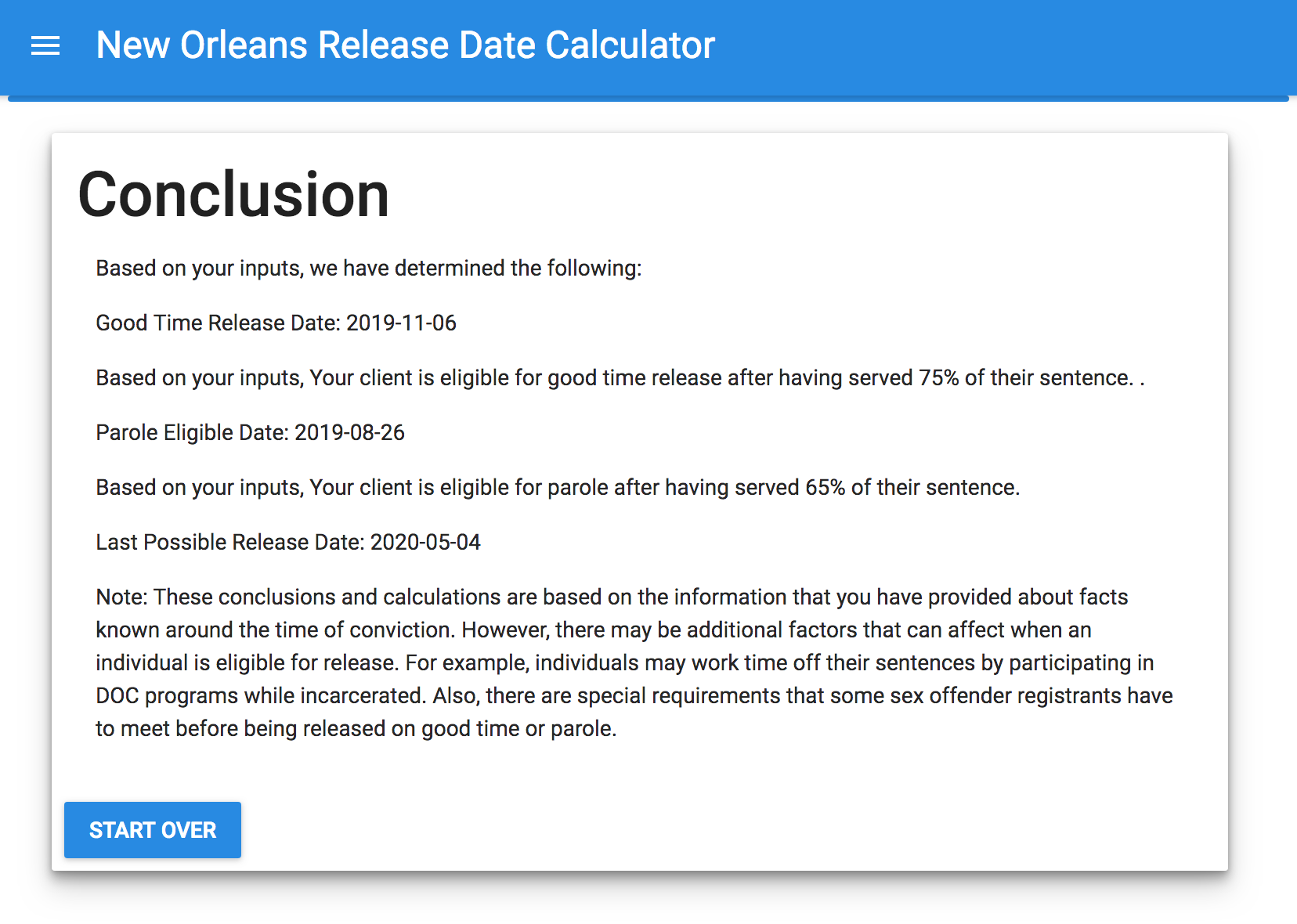
7. Date of client’s arrest and conviction

****

7. (Date selection from calendar)



8. Final Report and Conclusions



**Lessons Learned:**

**This is hard.** Even for those students with a strong base in python, creating an application in this way will be significantly challenging. As someone who studied computer science during my undergraduate program and who currently pursuing a computer science master’s, I still found this project to be extremely challenging. Learning how to program a front-end could likely fill a semester long course by itself. The back-end, while more straightforward for someone with python experience, is complicated by its need to receive data from and pass data back to the front-end.

**A comparison to building in Neota: What’s better, What’s worse**

Complex conditionals are much easier to program in python than in Neota and, for more sophisticated builders, programming applications in python will allow for much greater customization. The simplicity gains, however, end there. While an application, once planned, can be built in Neota by those with little to no programming experience in a few days, this will likely take students several weeks if not months to learn and program.

**Consideration of a python track for the Iron Tech competition**

A python track will require a much greater time commitment which may not be feasible given how many changes are requested by clients late in the semester.

If a python track were to be added to the Iron Tech course/competition, considerable instruction would likely be necessary. It might be wise to consider building templates of the front-end and back-end that students could modify for their needs.

**Personal learnings: I built an app!** My programming experience has centered on back-end development so this was new and very challenging. I am thrilled to have built a full application that replicates the basic functionality of the New Orleans Release Date Calculator built for the Iron Tech competition and hope to continue to expand my skills by building more applications over the summer.