**PRUEBA TÉCNICA**

**Descripción de la arquitectura utilizada:**

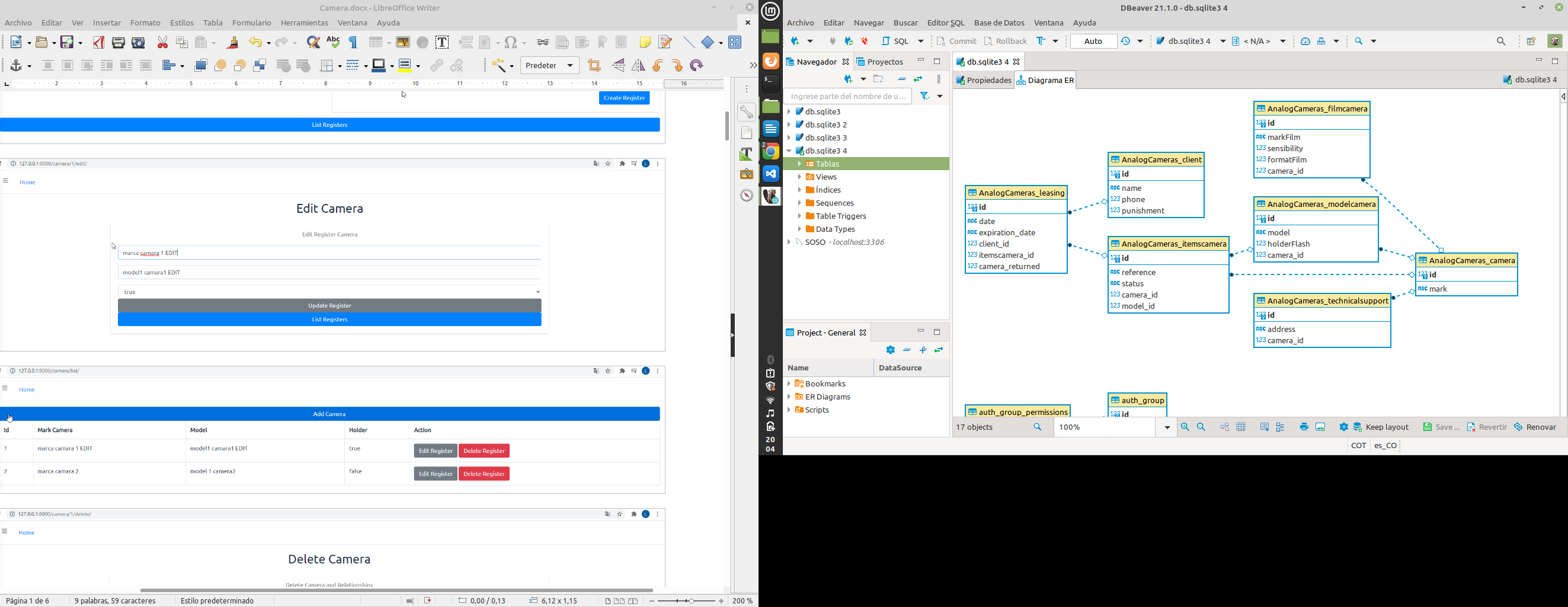
La metodología implementada para el sistema de información propuesto, se realiza a través del desarrollo web; utilizando del lado del servidor (BACKEND) un framework llamado **Django** y en el lado del usuario (FRONTEND) se incorpora **Vuejs** como framework . Cabe mencionar que los frameworks mencionados anteriormente son de código abierto, es decir *open source*

**Django** es un [framework](https://es.wikipedia.org/wiki/Framework) de desarrollo web escrito en [Python](https://es.wikipedia.org/wiki/Python), que respeta el patrón de diseño conocido como M[odelo–Vista–Controlador](https://es.wikipedia.org/wiki/Modelo–vista–controlador) (MVC). Las caracterizara mas sobresaliente de Django es la posibilidad de crear aplicaciones reinstalables con el fin de aplicar el principio “No te repitas”,(DRY, del inglés *Don't Repeat Yourself*) y de esta manera reducir el código de ejecución.

A su vez Django posee un sistema de vista genéricas lo cual facilita la elaboración de API REST

Se propone una base de datos relacional y como gestor de base de datos se lleva a cabo por medio de Sqlite, que a su vez también es *open source* y se halla de manera predeterminada por Django para el modelo de negocios de la aplicación.

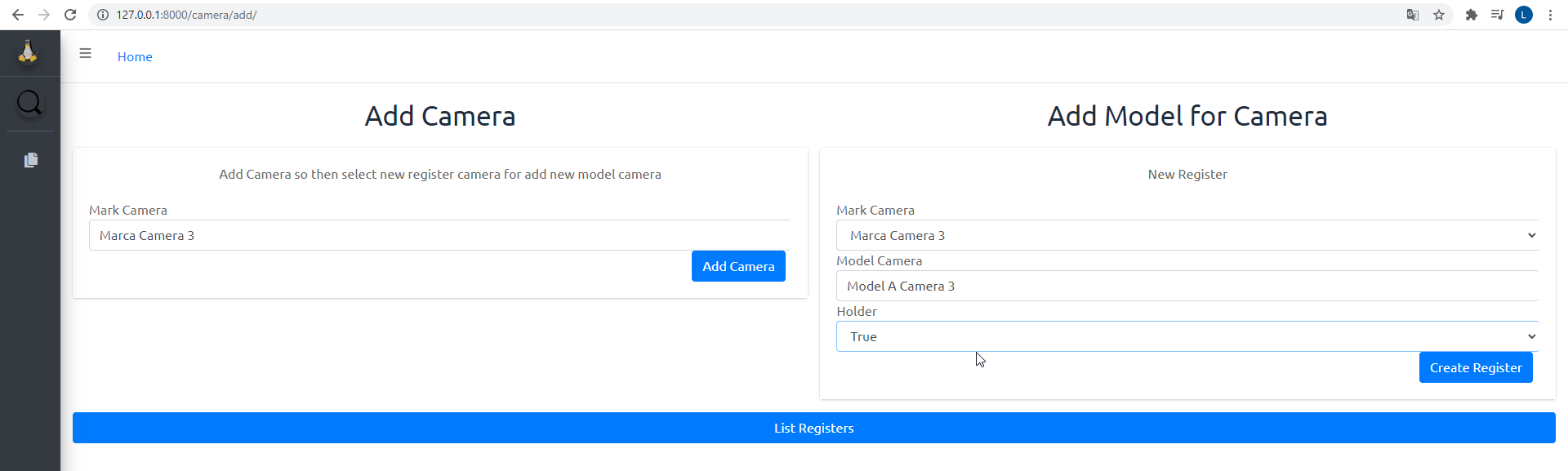
La esquema de la relación de tablas de la base de datos, se muestra a continuación:

Figura 1: Esquema Base de Datos Relacional

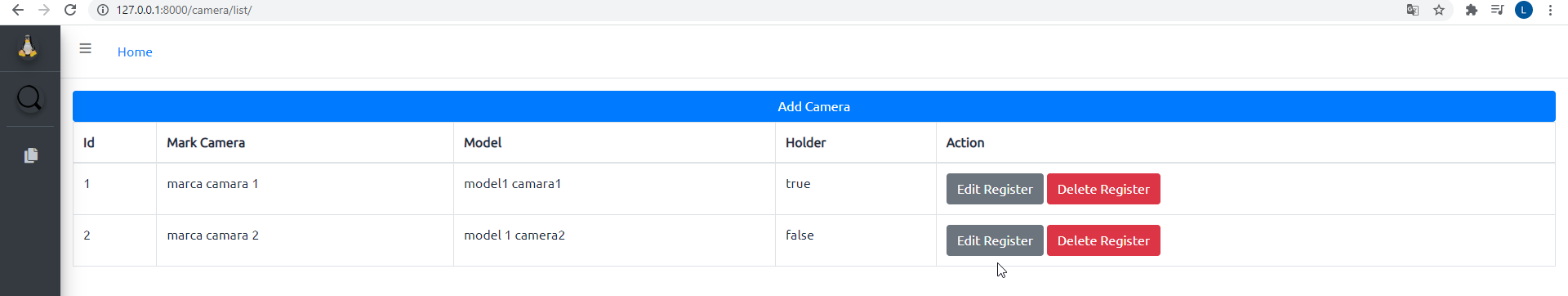
Por otra parte,**Vuejs** es un framework *open source* de **JavaScript**, el cual nos permite construir interfaces de usuarios de manera simple. Para este caso especifico la reactividad que ofrece Vue es de gran ayuda, es decir el cambio de estado entre objectos se hace de manera interna sin recargar la pagina, debido a que dichos objectos son “monitoreados” por el framework ante posibles cambios.

A continuación se presentan los resultados obtenidos en cada de las pruebas realizadas, teniendo en cuenta que cada modelo relacionado en la base de datos posee su respectivo CRUD:

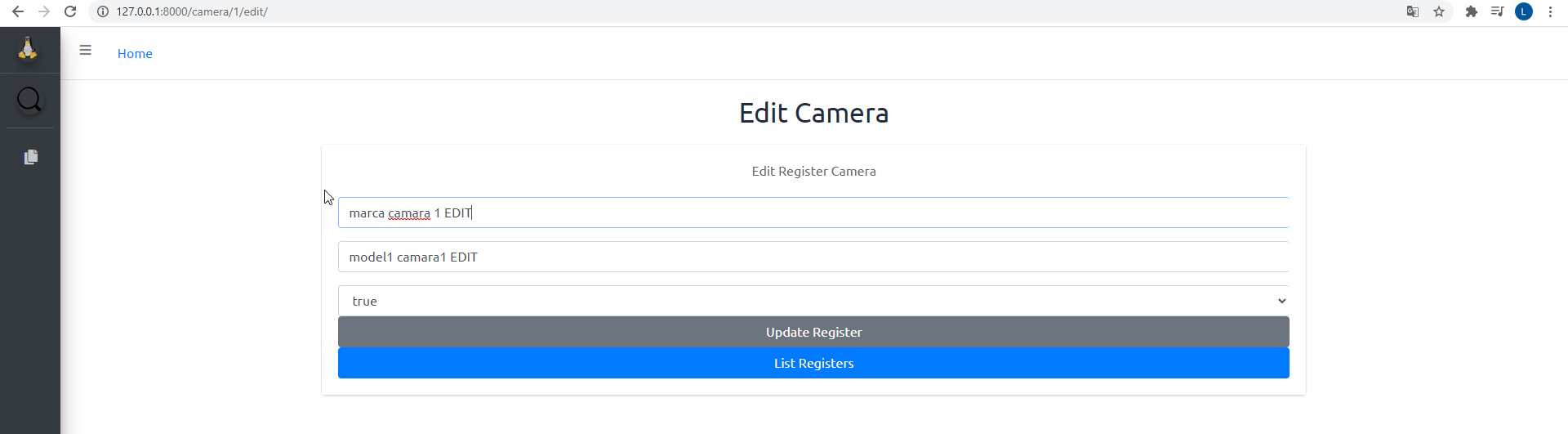
Camera **C**reate

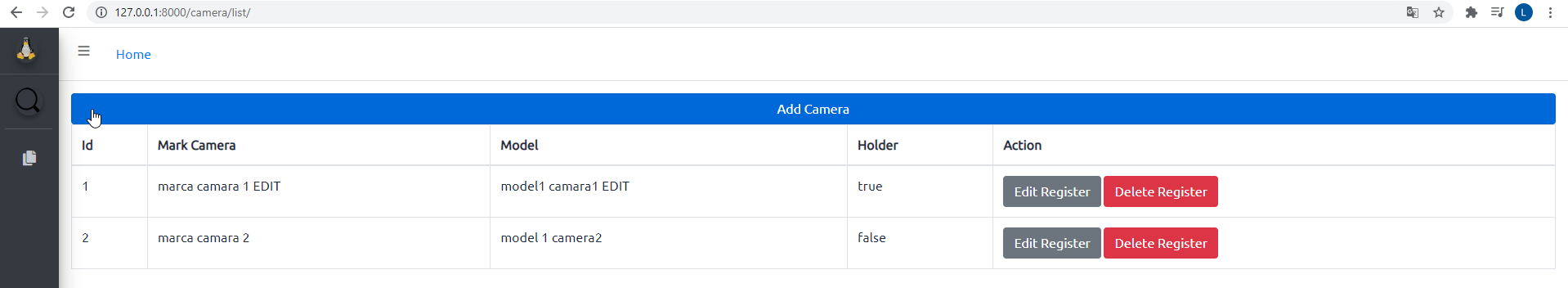


Camera **R**etrieve

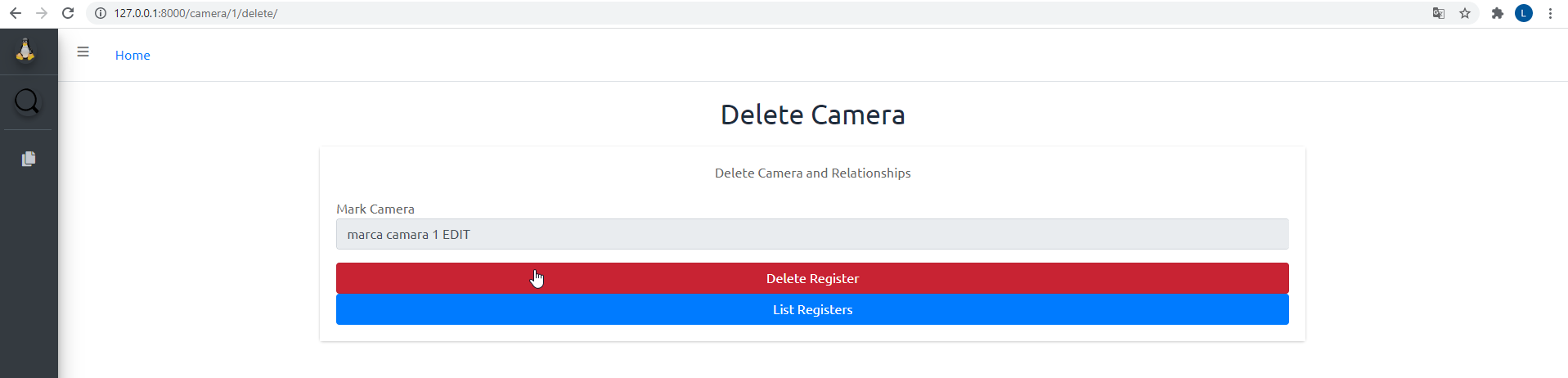


Camera **U**pdate

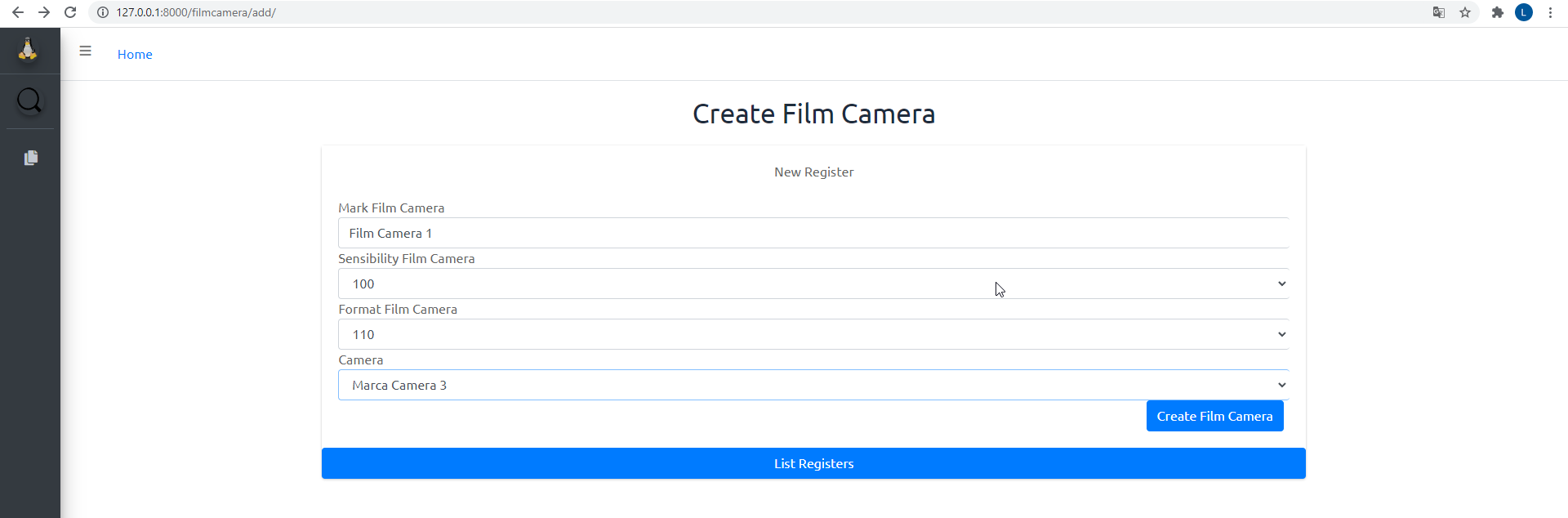




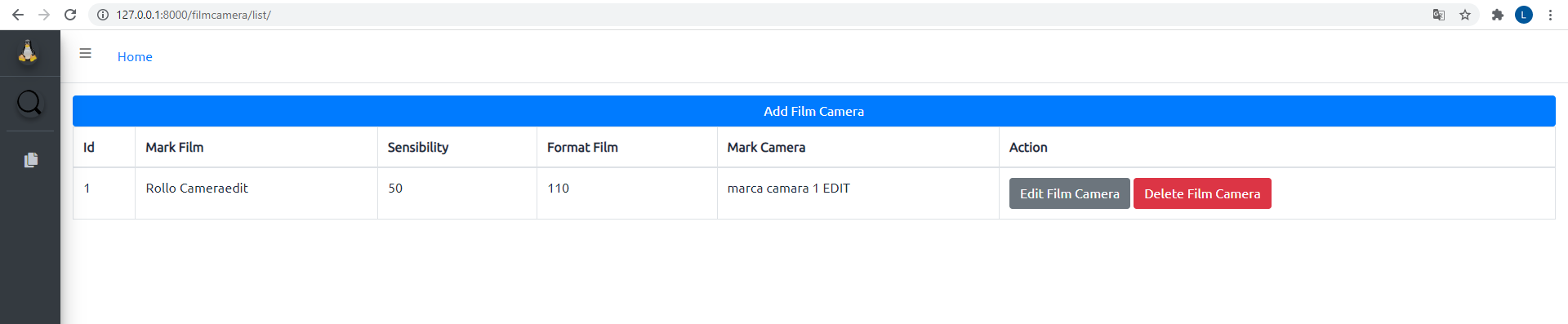
Camera **D**elete



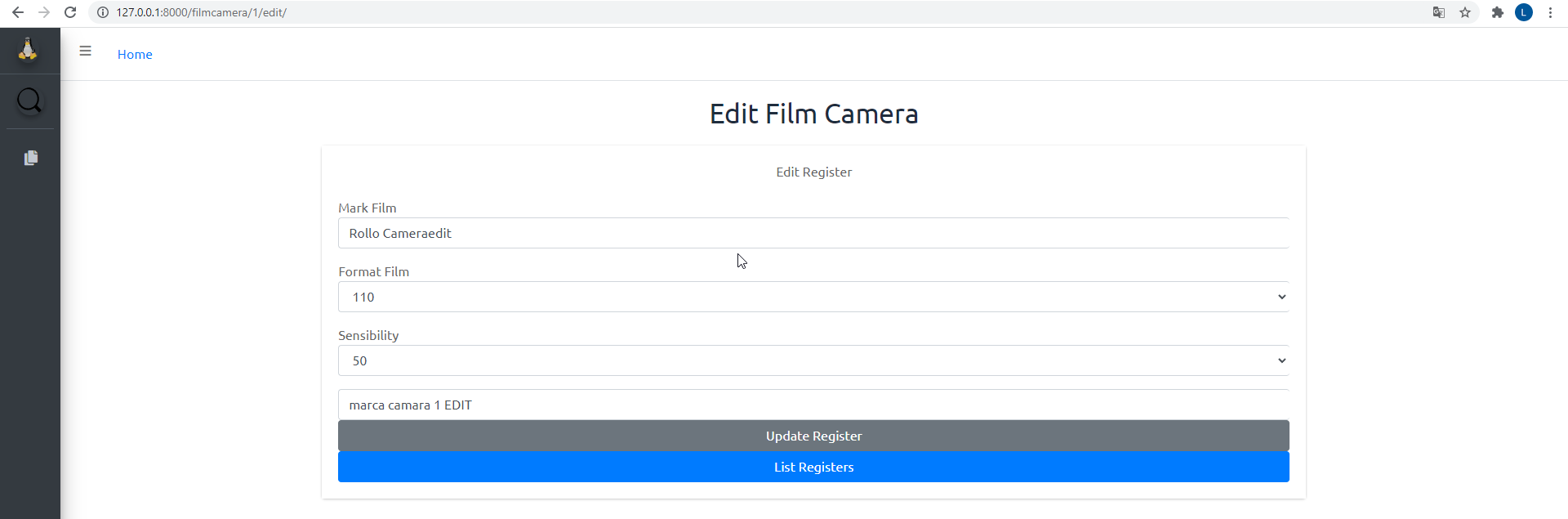
Film Camera **C**reate



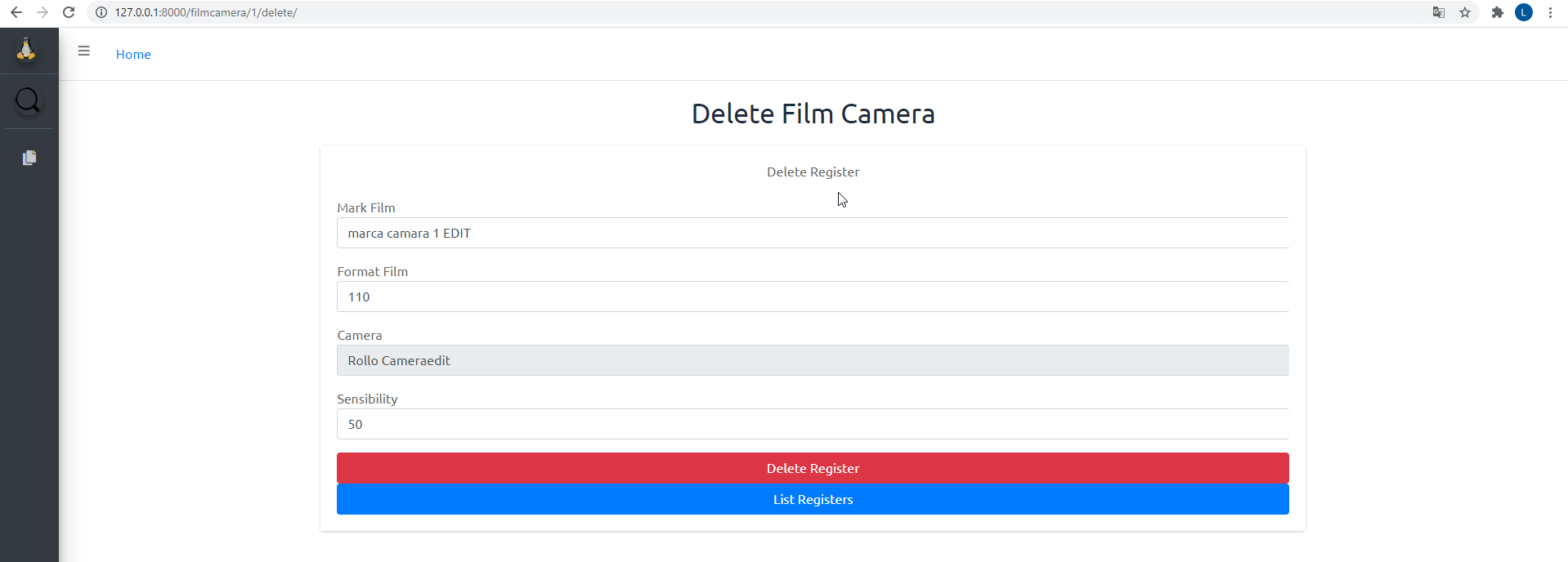
Film Camera **R**etrieve



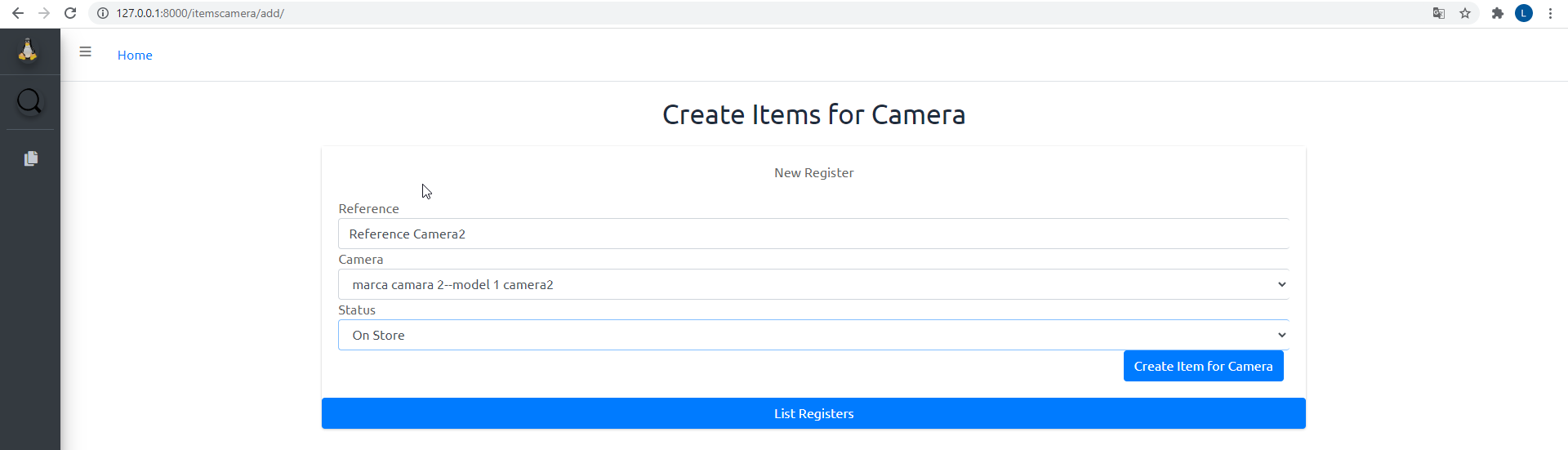
Film Camera **U**pdate



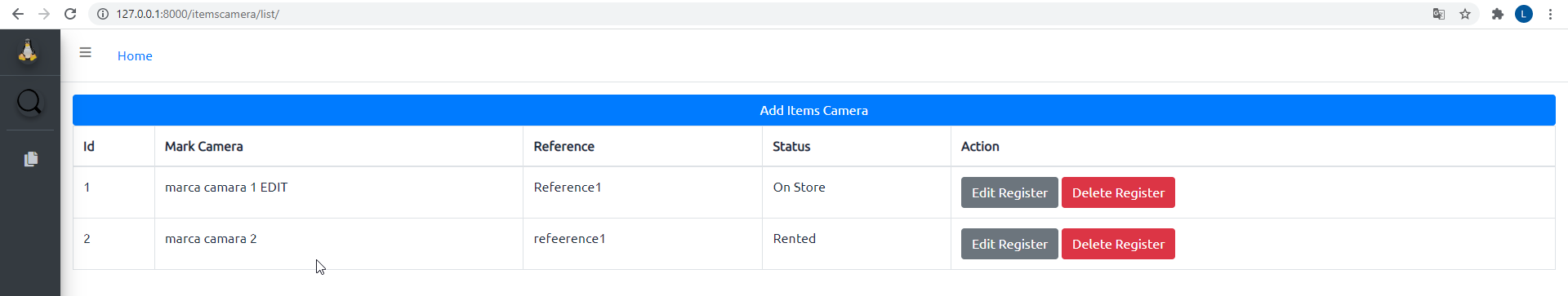
Film Camera **D**elete



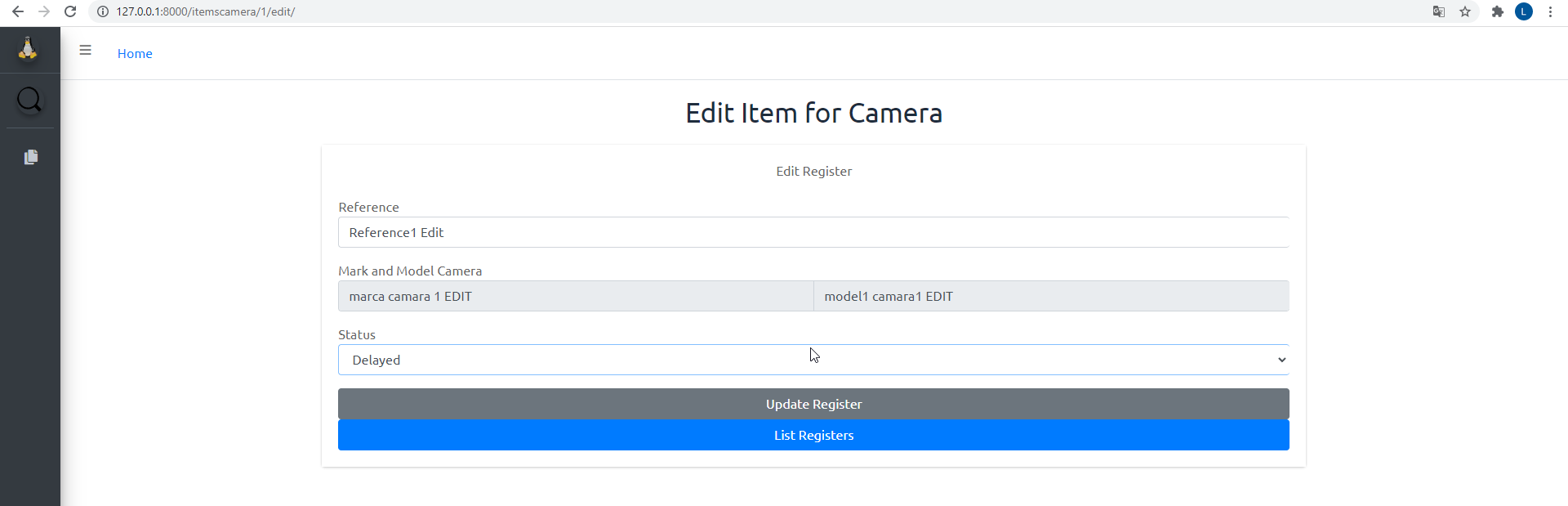
Items Camera **C**reate

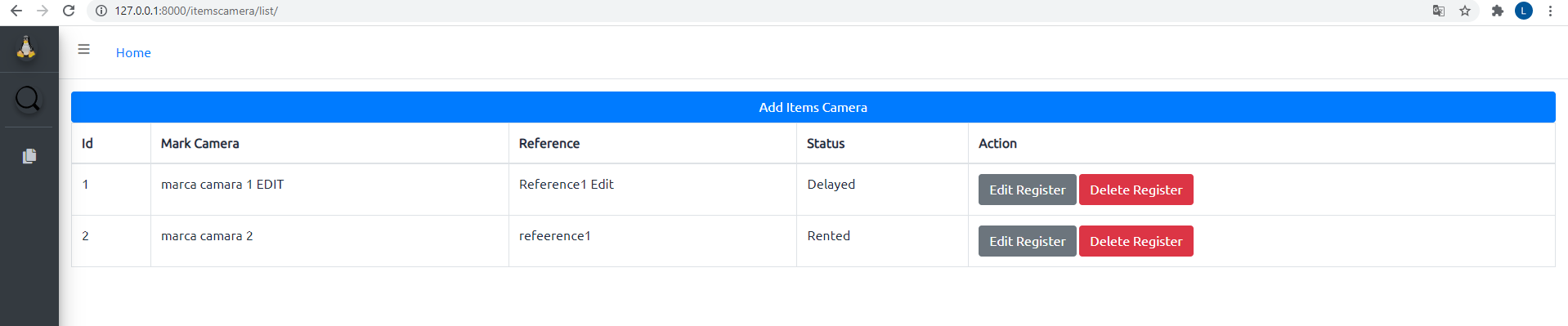


Items Camera **R**etrieve

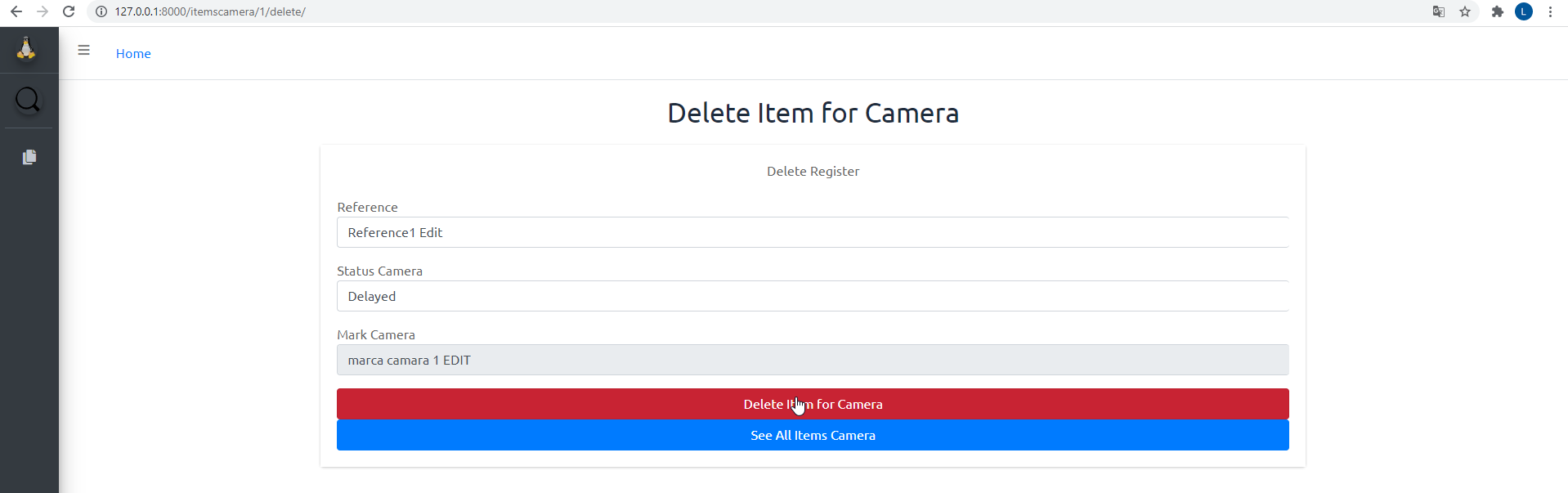


Items Camera **U**pdate

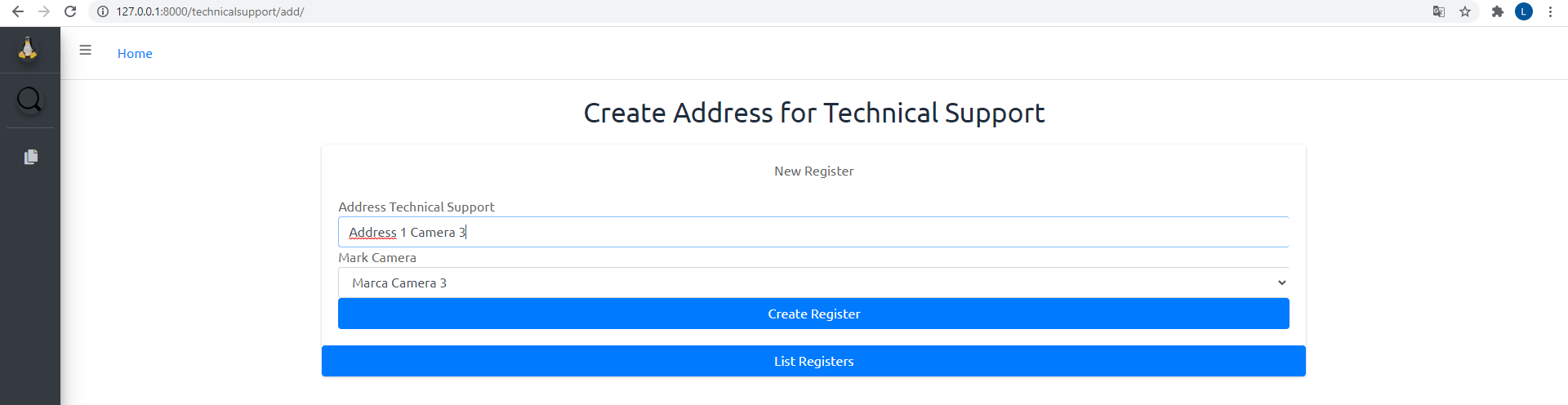




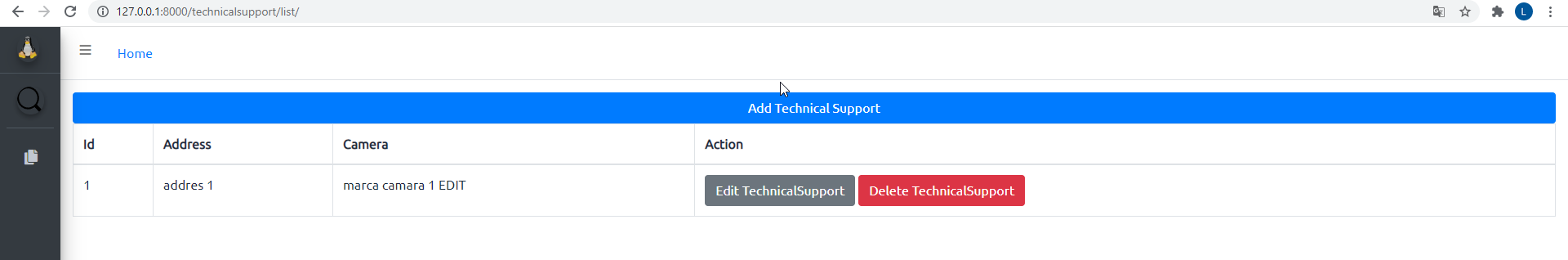
Items Camera **D**elete



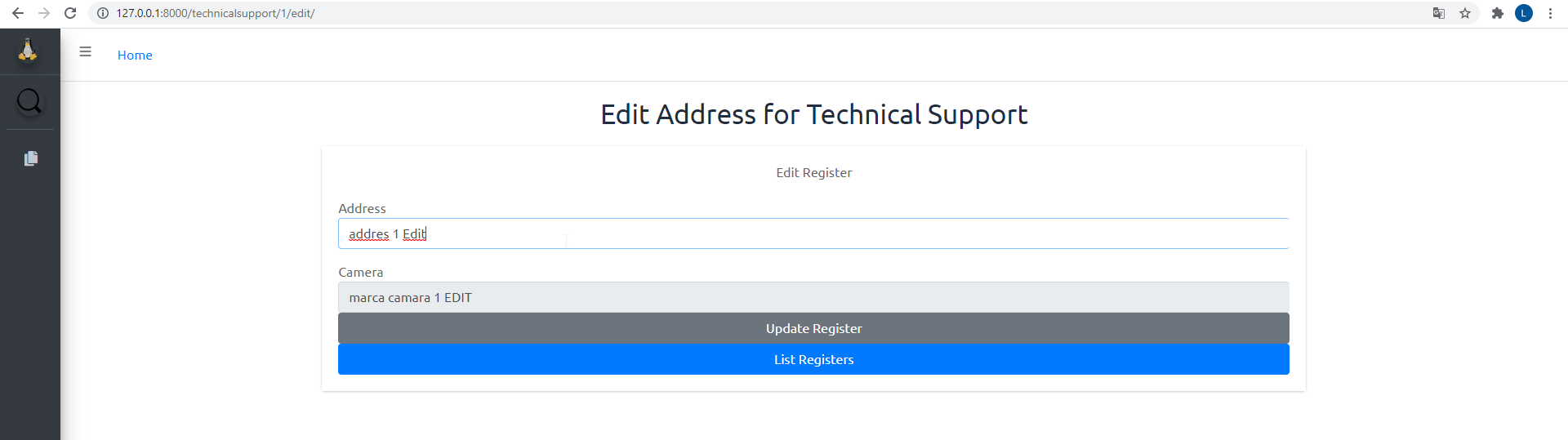
Technical Support **C**reate



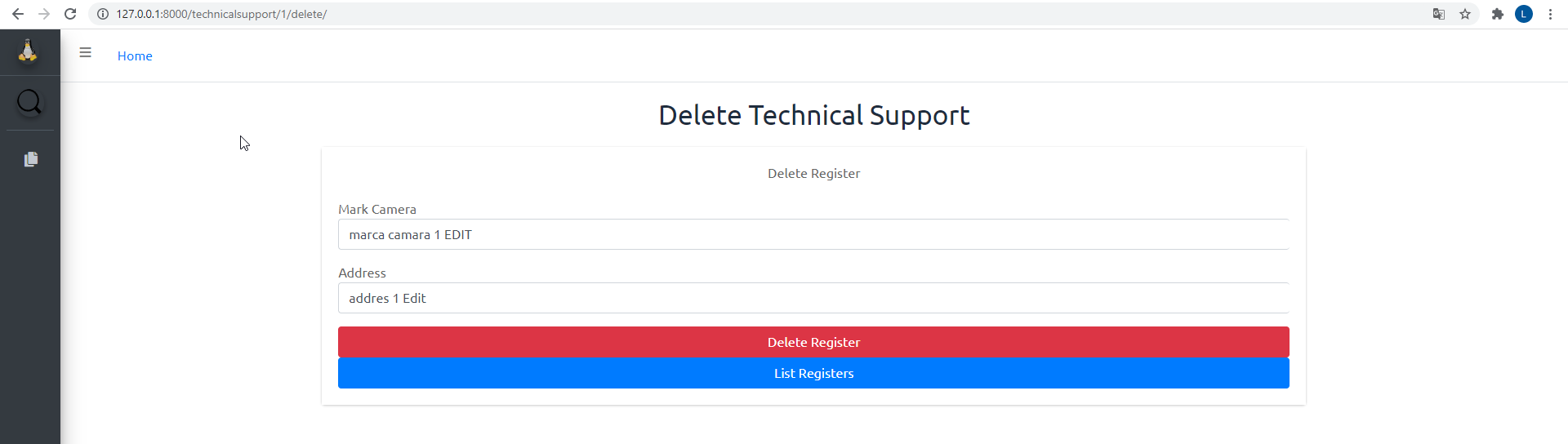
Technical Support **R**etrieve



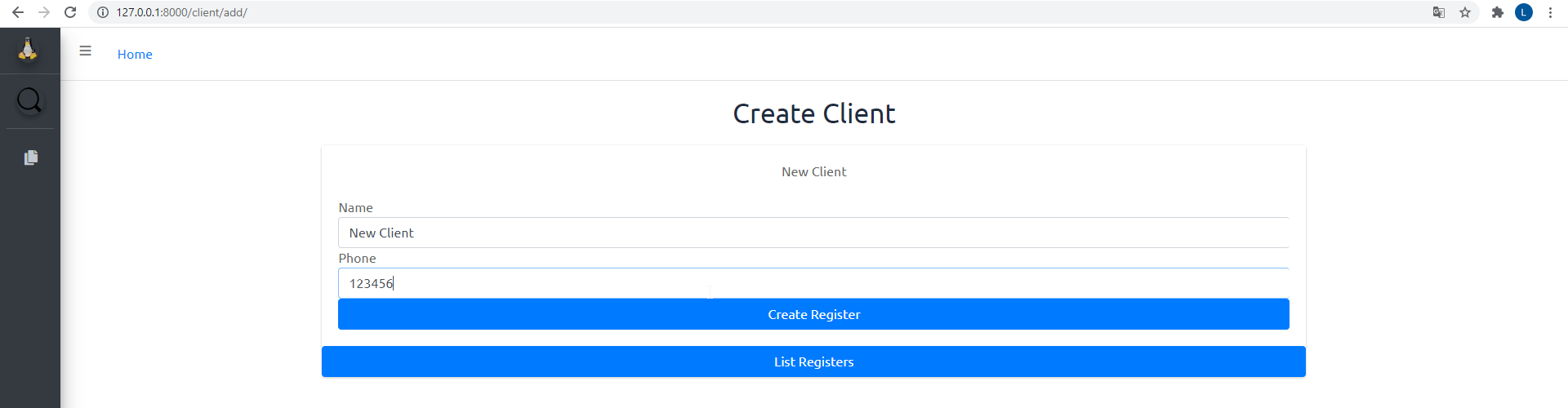
Technical Support **U**pdate



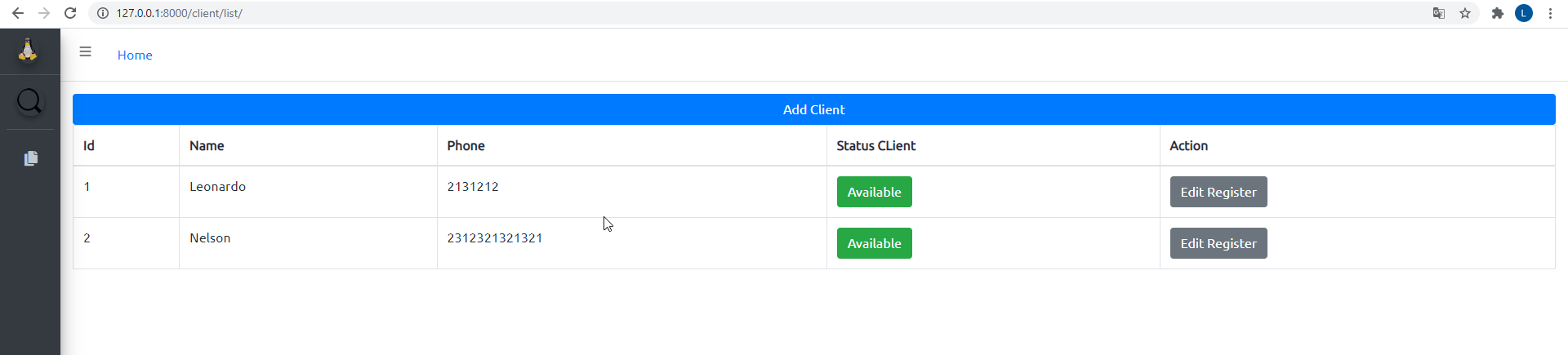
Technical Support **D**elete



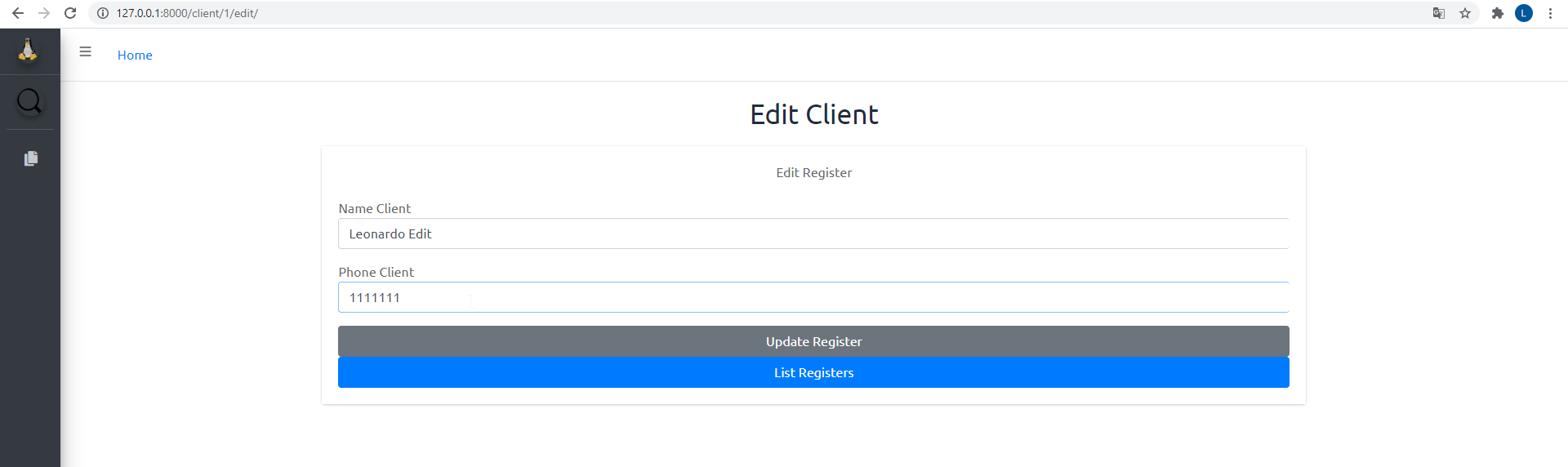
Client **C**reate

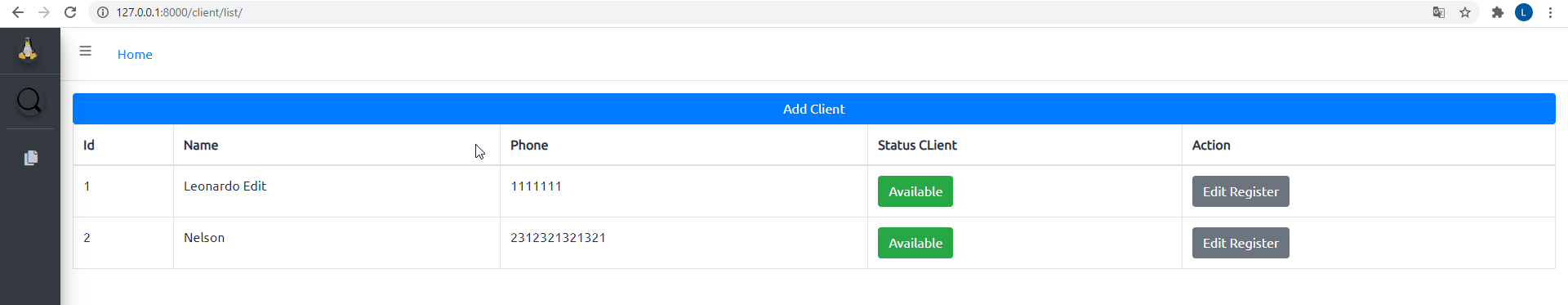


Client **R**etrieve



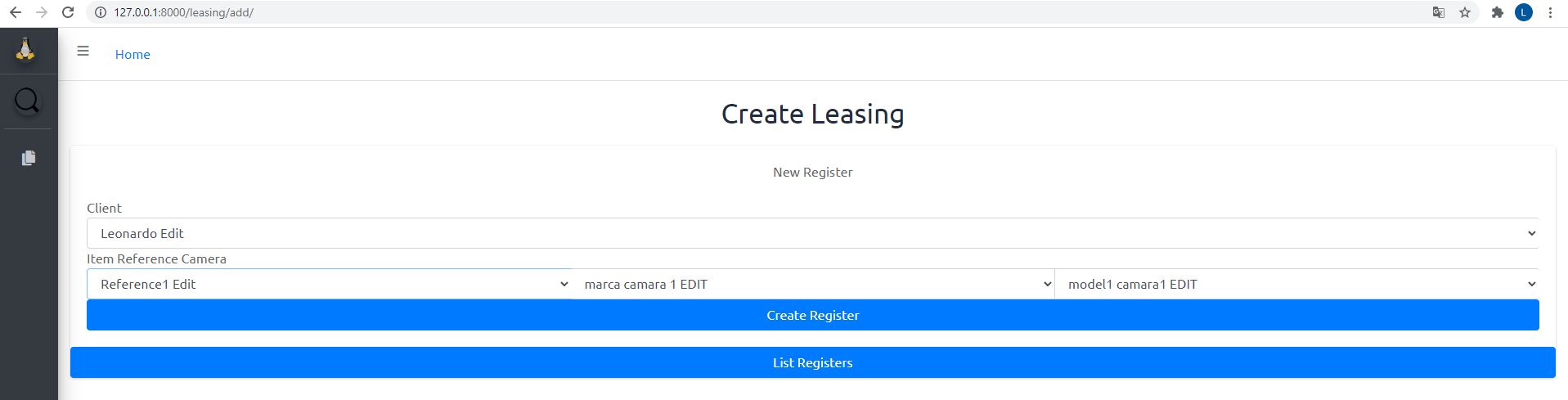
Client **U**pdate



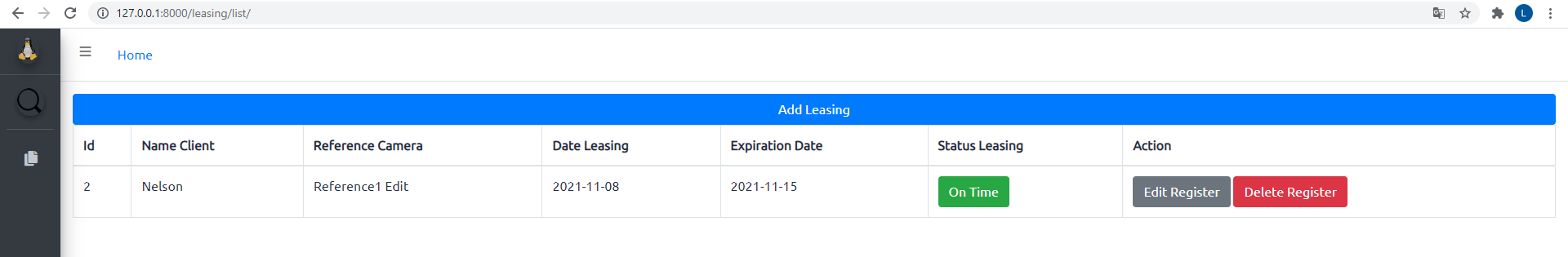


Client **D**elete : No se efectuó eliminación de clientes, ya que de esta manera se logra llevar registro de la cantidad de clientes que visitan la tienda de cámaras.

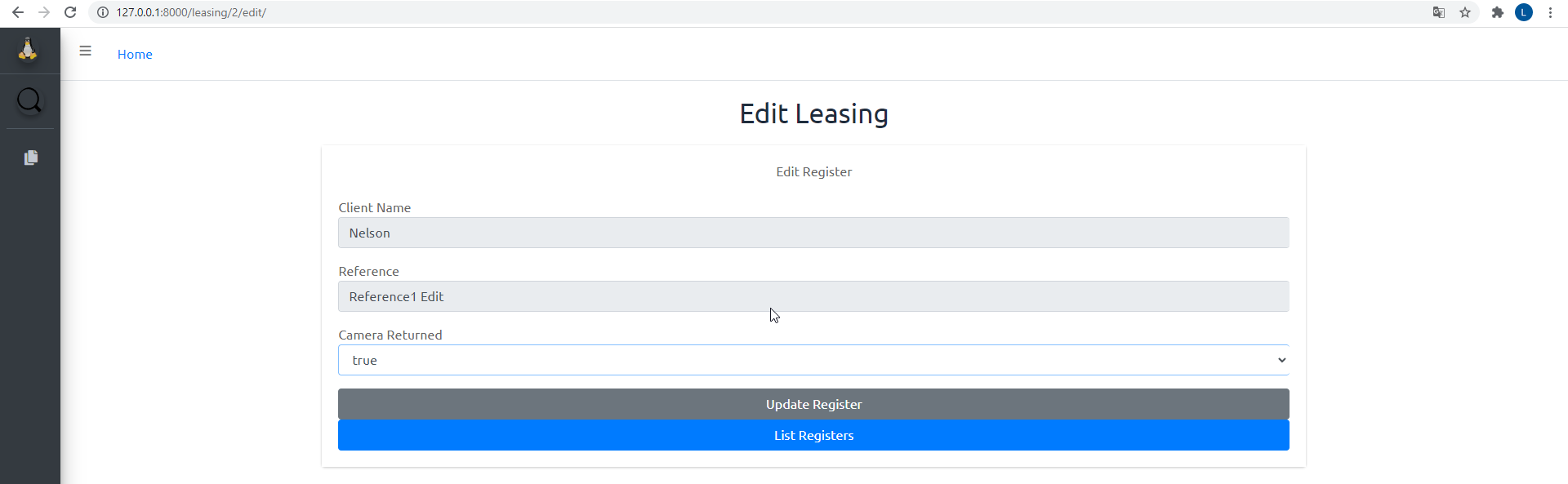
Leasing **C**reate



Leasing **R**etrieve



Leasing **U**pdate



Leasing **D**elete

