The task was to implement a distributed file system. The whole programming was done in python, using the web.py framework extensively. Persistent storage was implemented for most of the services using the library shelve. Other notable libraries which were used include – sys, os, requests, HTTPBasicAuth, re and base64.

Across all services, shelve was used to store values in in a (key, object) manner. Web.py was used to achieve RESTful architecture, and effortless communication using GET and POST.

The following features were attempted - Distributed Transparent File Access, Security Service, Directory Service and Lock Service. Here is a short description of each of these services:

Distributed Transparent File Access:

The program implementing this feature is `fileserver.py`. Fileserver is a RESTful service that takes the filename using through URL and perform an appropriate action on it (Write/Read). This can take port number as an argument and run on different ports simultaneously.

Security Service

A login service using HTTPBasicAuth was implemented for this service. The program implementing this feature is `authserver.py.` The program features two classes – 'login' and 'newuser'. RESTful calls are made between this program and others, to facilitate new user and existing users. A BASE64 encode/decode was implemented here.

Directory Service

The directory service implemented in directoryserver.py returns the filepath, including the port to a requested service. A different program directorystore.py was used to populate the database initially

Lock Service

To ensure data integrity, a lock service was implemented and called whenever a write operation was involved. This was also implemented as a RESTful service using web.py. The files involved are lockserver.py and lockstore.py.