```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "outputs": [],
   "source": [
    "from pathlib import Path\n",
    "import json\n",
    "import os\n",
    "\n",
    "import pandas as pd\n",
    "from tinydb import TinyDB\n",
    "\n",
    "current_dir = Path(os.getcwd()).absolute()\n",
    "results_dir = current_dir.joinpath('results')\n
  ",
    "kv_data_dir = results_dir.joinpath('kvdb')\n",
    "kv_data_dir.mkdir(parents=True, exist_ok=True)"
   ],
   "metadata": {
    "collapsed": false,
    "pycharm": {
     "name": "#%%\n"
    }
   }
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "outputs": [
    {
     "data": {
      "text/plain": "4"
     },
     "execution_count": 2,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
```

```
    "people_json = kv_data_dir.joinpath('people.json')\n",
    "visited_json = kv_data_dir.joinpath('visited.json')\n",
    "sites_json = kv_data_dir.joinpath('sites.json')\n",
    "measurements_json = kv_data_dir.joinpath('measurements.json')\n",
    "\n",
    "\n",
    "paths = [\n",
    "    people_json,\n",
    "    visited_json,\n",
    "    sites_json,\n",
    "    measurements_json\n",
    "]\n",
    "\n",
    "contents = []\n",
    "\n",
    "for path in paths:\n",
    "    with open(path, 'r') as f:\n",
    "        contents.append(json.loads(f.read()))\n",
    "\n",
    "# print(type(json.dumps(contents)))\n",
    "\n",
    "\n",
    "\n",
    "db_json = json.dumps(contents)\n",
    "type(contents[0])\n",
    "len(contents)"
   ],
   "metadata": {
    "collapsed": false,
    "pycharm": {
     "name": "#%%\n"
    }
   }
  },
  {
   "cell_type": "code",
```

```
 77    "execution_count": 3,
 78    "metadata": {
 79     "pycharm": {
 80      "name": "#%%\n"
 81     }
 82    },
 83    "outputs": [],
 84    "source": [
 85     "class DocumentDB(object):\n",
 86     "    def __init__(self, db_path):\n",
 87     "        ## You can use the code from the
    previous example if you would like\n",
 88     "        people_json = kv_data_dir.joinpath('
    people.json')\n",
 89     "        visited_json = kv_data_dir.joinpath('
    visited.json')\n",
 90     "        sites_json = kv_data_dir.joinpath('
    sites.json')\n",
 91     "        measurements_json = kv_data_dir.
    joinpath('measurements.json')\n",
 92     "\n",
 93     "        self._db_path = Path(db_path)\n",
 94     "        self._db = None\n",
 95     "        ## TODO: Implement code\n",
 96     "\n",
 97     "        paths = [\n",
 98     "            people_json,\n",
 99     "            visited_json,\n",
100     "            sites_json,\n",
101     "            measurements_json\n",
102     "        ]\n",
103     "\n",
104     "        contents = []\n",
105     "\n",
106     "        for path in paths:\n",
107     "            with open(path, 'r') as f:\n",
108     "                contents.append(json.loads(f.
    read()))\n",
109     "\n",
110     "        self._db_json = contents\n",
111     "        self._load_db()\n",
```

```
    "\n",
    "\n",
    "    def _load_db(self):\n",
    "        self._db = TinyDB(self._db_path)\n",
    "        ## TODO: Implement code\n",
    "        # self._db.insert(self._db_json)\n",
    "\n",
    "\n",
    "        for x in range(len(self._db_json)):\n",
    "            self._db.insert(self._db_json[x])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "outputs": [],
   "source": [
    "db_path = results_dir.joinpath('patient-info.
json')\n",
    "if db_path.exists():\n",
    "    os.remove(db_path)\n",
    "\n",
    "db = DocumentDB(db_path)"
   ],
   "metadata": {
    "collapsed": false,
    "pycharm": {
     "name": "#%%\n",
     "is_executing": true
    }
   }
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
```

```
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.8.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 4
}
```