A BRIEF EXPLANATION OF THE IMPLEMENTATION OF THE GET ENDPOINTS
by Michael Mortty

Listing of Users

```
app.get('/users', (req, res) => {
  const users = getUsersData();
  res.json(users);
});
```

With the help of the `getUsersData()` function within the code, the function is able to read the users.json files and is able to return all of them. The res.json(users) is used to send whatever data is within the file as JSON

Getting User IDs

```
app.get('/users/:id', (req, res) => {
  const users = getUsersData();
  const userId = req.params.id;
  const user = Object.values(users).find(u => u.id === parseInt(userId));
  if (user) {
    res.json(user);
  } else {
    res.status(404).send('No users found');
  }
});
```

This also uses the `getUsersData()` function with the added use of rq.params.id, which extracts the id of the user from the JSON file. The find method being used searches for a user whose id matches the parsed value of the id parameter. Should a match be found the user is returned as a JSON response, else a 404 error with the message “No users found” is sent.

Getting User Professions

```
app.get('/users/profession/:profession', (req, res) => {
  const users = getUsersData();
  const profession = req.params.profession.toLowerCase();
  const filteredUsers = Object.values(users).filter(
    u => u.profession.toLowerCase() === profession
  );
  if (filteredUsers.length > 0) {
    res.json(filteredUsers);
  } else {
    res.status(404).send('No users found');
  }
});
```

Again the `getUsersData()` function is used but this time the profession parameter to be extraced from the URL is converted to lowercase for a case sensitive search. The filter method is used to collect users whose profession matches the parameter. Should a match be found the user is returned as a JSON response, else a 404 error with the message “No users found” is sent.

Getting User Names

```
app.get('/users/name/:name', (req, res) => {
  const users = getUsersData();
  const name = req.params.name.toLowerCase();
  const user = Object.values(users).find(
    u => u.name.toLowerCase() === name
  );
  if (user) {
    res.json(user);
  } else {
    res.status(404).send('No users found');
  }
});
```

With the use of `getUsersData()` function and the find method, this searches for a user, within the JSON file, whose name matches the parameter and then retrieves it. Should a match be found the user is returned as a JSON response, else a 404 error with the message “No users found” is sent. It’s also converted into lowercase for a case sensitive search.

This code wouldn’t have been possible without the help of a few classmates and a very benevolent senior.