

CS446 Assignment3

I. Requirements

By utilizing Python language, and relevant libraries please create a web service code `Lastname_Assignemnt3.py`. Your code shall implement following functionalities.

1. Shall implement a webservice and connect to mongodb server. The mongodb server shall have “cs446” database and “oscars” collection, which is populated with the contents of “oscar_age_female.csv” and “oscar_age_male.csv” files.

2. Shall implement the following “post” method.

<http://<servername>:<port>/populate>

Example:

When a client sends the following json data with a post method, the server shall store the data in the “oscars” collection.

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",
  "Name": "Renee Zellweger",
  "Movie": "What/If"
}
```

3. Shall implement the following “get” method.

<http://<servername>:<port>/index>

Example:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{
  "Index": "999"
}
```

Response:

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",
  "Name": "Renee Zellweger",
  "Movie": "What/If"
}
```

4. Shall implement the following “get” method.

<http://<servername>:<port>/year>

Example:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{
  "Year": "2020"
}
```

Response:

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",
  "Name": "Renee Zellweger",
  "Movie": "What/If"
}
```

5. Shall implement the following “get” method.

<http://<servername>:<port>/age>

Example:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{
  "Age": "51"
}
```

Response:

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",
  "Name": "Renee Zellweger",
  "Movie": "What/If"
}
```

6. Shall implement the following “GET” method.

<http://<servername>:<port>/name>

Example 1:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{
  "Name": "Zellweger"
}
```

Response:

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",
  "Name": "Renee Zellweger",
  "Movie": "What/If"
}
```

Example 2:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{
  "Name": "Renee "
}
```

Response:

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",
  "Name": "Renee Zellweger",
  "Movie": "What/If"
}
```

7. Shall implement the following “GET” method.

<http://<servername>:<port>/movie>

Example 1:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{
  "Movie": "What"
}
```

Response:

```
{
  "Index": "999",
  "Year": "2020",
  "Age": "51",

```

```
"Name": "Renee Zellweger",  
"Movie": "What/If"  
}
```

Example 2:

When a client sends the following json data with the “GET” method, the server shall search the data in the “oscars” collection and respond with correct result:

Request:

```
{  
  "Movie": "If"  
}
```

Response:

```
{  
  "Index": "999",  
  "Year": "2020",  
  "Age": "51",  
  "Name": "Renee Zellweger",  
  "Movie": "What/If"  
}
```

8. Shall implement a “GET” method for the following URL.

<http://<servername>:<port>/movie/<string:if>>

Example:

When a client tried to open the url <http://<servername>:<port>/movie/if> , the server shall search the data in the “oscars” collection and respond with correct result:

Response:

```
{  
  "Index": "999",  
  "Year": "2020",  
  "Age": "51",  
  "Name": "Renee Zellweger",  
  "Movie": "What/If"  
}
```