

PROJECT IMPLEMENTATION

1. Run following commands in command prompt (cmd) :

```
$ flask init  
$ foreman start
```

\$ flask init: With this command, flask starts and database is created newly
\$ foreman start : This command helps to run 2 separate Flask applications connected to a single SQLite Version 3 database.

2. Users microservice

- **createUser(username, email, password)**

This is a “POST” method :- **@app.route('/user/create, methods=['POST'])**

Command to use:

```
$ curl localhost:5000/user/create -d '{"username": "Harshada", "email":  
"harshada@gmail.com", "password": "1234"}' -H 'Content-Type: application/json'
```

Output:

```
{  
  "Message": "User created succesfully",  
  "content-language": "en-US",  
  "content-type": "application/json",  
  "status code": "201 - Created"  
}
```

- **authenticateUser(username, password)**

Returns true if the supplied password matches the hashed password stored for that username in the database.

This is a “POST” method :- **@app.route('/user/authenticateUser', methods=['POST'])**

Command to use:

```
$ curl localhost:5000/user/authenticateUser -d '{"username": "Harshada", "password": "1234"}' -H 'Content-Type: application/json'
```

Output:

```
{
  "Message": "User authenticated successfully",
  "content-language": "en-US",
  "content-type": "application/json",
  "status code": "200 - OK"
}
```

- **addFollower(username, usernameToFollow)**

This is a POST request :- `@app.route('/user/addFollower', methods=['POST'])`

Command to use:

```
$ curl localhost:5000/user/addFollower -d '{"username": "Harshada", "usernameToFollow": "Rahul"}' -H 'Content-Type: application/json'
```

Output:

```
{
  "Message": "User follower created succesfully",
  "content-language": "en-US",
  "content-type": "application/json",
  "status code": "201 - Created"
}
```

- **removeFollower(username, usernameToRemove)**

@app.route('/user/removeFollower', methods=['DELETE'])

Command to use:

```
$ curl -i -X DELETE -H 'Content-Type:application/json' -d '{"username": "Harshada", "usernameToUnfollow": "Pooja"}' localhost:5000/user/removeFollower
```

Output:

```
{
  "Message": "Follower removed succesfully",
  "content-language": "en-US",
  "content-type": "application/json",
  "status code": "200 - OK"
}
```

3. Timelines microservice

- **getUserTimeline(username)**

This is a GET Method:- @app.route('/<username>/userTimeline', methods=['GET'])

Command to use:

```
$ curl localhost:5100/user/Harshada/userTimeline
```

Output:

```
[
  {
    "createDate": "2020-10-08 23:04:39.704811",
    "tweet": "Harshada: Eating is good for health",
    "username": "Harshada"
  },
]
```

```
{
  "createDate": "2020-10-08 23:04:16.779375",
  "tweet": "Pooja: Eating is good for health",
  "username": "Harshada"
},
{
  "createDate": "2020-10-08 23:01:18.149641",
  "tweet": "Hi foreman is running properly",
  "username": "Harshada"
}
]
```

Screenshot:



```
0:
  createDate: "2020-10-08 23:04:39.704811"
  tweet: "Harshada: Eating is good for health"
  username: "Harshada"
1:
  createDate: "2020-10-08 23:04:16.779375"
  tweet: "Pooja: Eating is good for health"
  username: "Harshada"
2:
  createDate: "2020-10-08 23:01:18.149641"
  tweet: "Hi foreman is running properly"
  username: "Harshada"
```

- **getPublicTimeline()**

This is a GET method:- `@app.route('/user/publicTimeline', methods=['GET'])`

Command to use:

```
$ curl localhost:5100/user/publicTimeline
```

Output:

```
[
  {
    "createDate": "2020-10-08 23:06:54.174353",
    "tweet": "Pooja12345",
    "username": "Pooja"
  },
  {
    "createDate": "2020-10-08 23:04:39.704811",
    "tweet": "Harshada: Eating is good for health",
    "username": "Harshada"
  },
  {
    "createDate": "2020-10-08 23:04:16.779375",
    "tweet": "Pooja: Eating is good for health",
    "username": "Harshada"
  },
  {
    "createDate": "2020-10-08 23:03:15.424883",
    "tweet": "Rahul: Yoga is good for health",
    "username": "Rahul"
  },
  {
    "createDate": "2020-10-08 23:02:41.769292",
    "tweet": "Hello World",
    "username": "Rahul"
  },
  {
    "createDate": "2020-10-08 23:01:18.149641",
    "tweet": "Hi foreman is running properly",
    "username": "Harshada"
  }
]
```

Screenshot:

```
▼ 0:
  createDate: "2020-10-08 23:06:54.174353"
  tweet:      "Pooja12345"
  username:   "Pooja"
▼ 1:
  createDate: "2020-10-08 23:04:39.704811"
  tweet:      "Harshada: Eating is good for health"
  username:   "Harshada"
▼ 2:
  createDate: "2020-10-08 23:04:16.779375"
  tweet:      "Pooja: Eating is good for health"
  username:   "Harshada"
▼ 3:
  createDate: "2020-10-08 23:03:15.424883"
  tweet:      "Rahul: Yoga is good for health"
  username:   "Rahul"
▼ 4:
  createDate: "2020-10-08 23:02:41.769292"
  tweet:      "Hello World"
  username:   "Rahul"
▼ 5:
  createDate: "2020-10-08 23:01:18.149641"
  tweet:      "Hi foreman is running properly"
  username:   "Harshada"
```

- **getHomeTimeline(username)**

@app.route('/user/<username>/homeTimeline', methods=['GET'])

Command to use:

```
curl localhost:5100/user/Harshada/homeTimeline
```

Output:

```
[
  {
    "createDate": "2020-10-08 23:03:15.424883",
    "tweet": "Rahul: Yoga is good for health",
    "username": "Rahul"
  },
  {
    "createDate": "2020-10-08 23:02:41.769292",
    "tweet": "Hello World",
    "username": "Rahul"
  }
]
```

Screenshot:



```
0:
  createDate: "2020-10-08 23:03:15.424883"
  tweet: "Rahul: Yoga is good for health"
  username: "Rahul"
1:
  createDate: "2020-10-08 23:02:41.769292"
  tweet: "Hello World"
  username: "Rahul"
```

- **postTweet(username, text)**

This is a POST method: **@app.route('/user/tweet', methods=['POST'])**

Command to use:

```
$ curl localhost:5100/user/tweet -d '{"username": "Harshada", "tweet": "Hi foreman is running properly"}' -H 'Content-Type: application/json'
```

Output:

```
{
  "Message": "Tweet created succesfully",
  "content-type": "application/json",
  "status code": "201 - Created"
}
```

