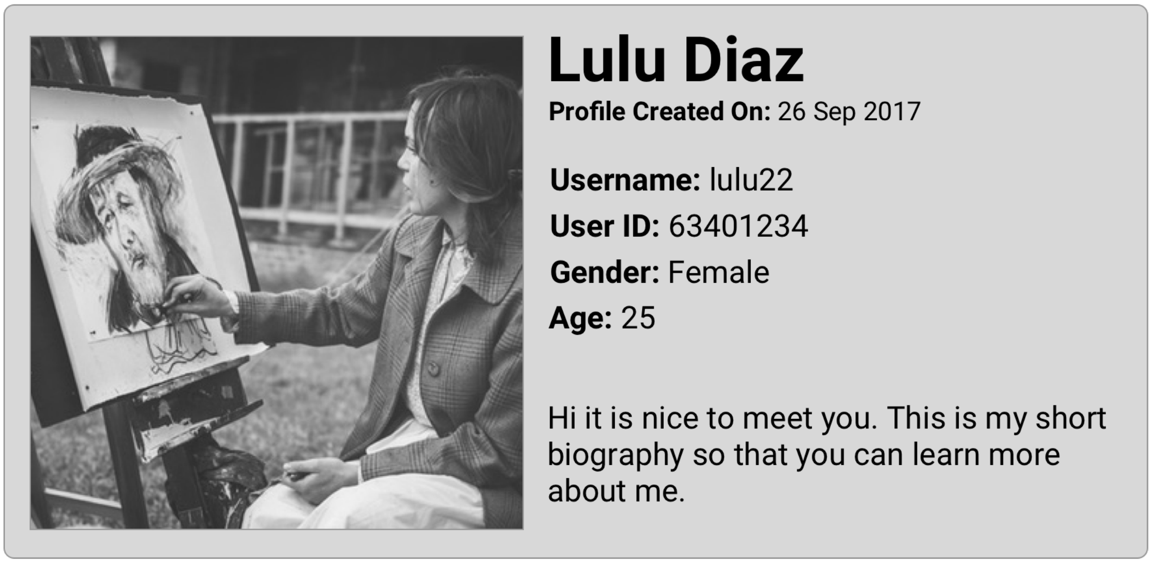
INFO3180 Project 1

Due: March 12, 2017

At the end of this project you will have a Flask based application that can accept and display profile information. The profile information will be stored in a PostgreSQL database. You will also begin to create an API for your application so that depending on the type of request that is made, you will return JSON data instead of rendering an HTML template.

Each individual profile will look like similar to the screenshot below in terms of the information that should be displayed. You are free to customize the design if you would like.



Lulu Diaz

Profile Created On: 26 Sep 2017

Username: lulu22

User ID: 63401234

Gender: Female

Age: 25

Hi it is nice to meet you. This is my short biography so that you can learn more about me.

Specifications

Part 1

Create a Flask App that accepts input for a user profile. The form must contain the following:

1. Text fields for **`firstname`**, **`lastname`**, **`age`**, **`biography`** (short)
2. File upload field called **`image`** which accepts the profile image
3. Select (option) field for **`gender`**

Upon submission, the form will validate the user input to prevent bad data and generate a unique **`userid`** and also generate the date the user was **`created_on`**. All of this input will be stored in a PostgreSQL database. Ensure that you create migrations for your user profile model.

The following routes will need to be created and an appropriate template rendered:

1. Name the route for adding a profile "**`/profile`**" **`GET`**
2. Name the route for viewing a list of all user profiles "**`/profiles`**" **`GET`**
3. Name the route for viewing an individual profile "**`/profile/<userid>`**" **`GET`**

Part 2

You will also begin the creation of a simple API for your user profiles. The API will use the following routes:

1. **`"/profiles" POST`**
2. **`"/profile/<userid>" POST`**

Each of these routes will return **JSON ONLY** instead of a rendered HTML template when a header of '**`Content-Type: application/json`**' sent with the request.

It should be possible to retrieve a list of all the users in **`JSON`** format from the system by sending a **`POST`** request along with the **`application/json`** Content-Type header to **`/profiles`**. (**Note:** ONLY Return the **username** and **userid**)

The result will be in JSON format:

```
{
"users":[
         {"username":"lulu22","userid":"63401234"},
         {"username":"bobbytables456","userid":"67453363"}, ...
       ]
}
```

It should also be possible to retrieve an individual user's profile information in **JSON** format from the system by sending a **POST** request along with the **application/json** Content-Type header to **/profile/<userid>**. (**Note:** DO NOT RETURN the first name and last name in the output).

The result will be in **JSON** format as shown below:

```
{
      "userid":"63401234",
      "username":"lulu22",
      "image":"myphoto.jpg",
      "gender": "Female",
      "age": "25",
      "profile_created_on":""
}
```

Note:

To generate your JSON output you might want to use the Flask **jsonify** method and to check the headers sent as part of the request you can use the **request.headers** dictionary. You can test your routes are working as expected by using the command line utility 'curl'. If you prefer a graphical interface you can use the Postman or Insomnia desktop app instead of curl.

Submission

Submit your code via the "Project 1 Submission" link on OurVLE. You should submit the following link:

1. Your Github repository URL for your Flask app e.g. https://github.com/{yourusername}/info3180-project1