\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* server.py \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

from flask\_app import app

from flask\_app.controllers import user\_controll

if \_\_name\_\_=="\_\_main\_\_":

app.run(debug=True, port=5004)

\*\*\*\*\*\*\*\*\*\*\*\*\* flask\_app/\_\_init\_\_.py \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

from flask import Flask

app=Flask(\_\_name\_\_)

app.secret\_key= 'hash my code !'

DATABASE='login\_and\_register'

\*\*\*\*\*\*\*\*\*\*\*\* flask\_app/config/mysqlconnection.py \*\*\*\*\*\*\*\*\*\*\*\*\*

# a cursor is the object we use to interact with the database

import pymysql.cursors

# this class will give us an instance of a connection to our database

class MySQLConnection:

def \_\_init\_\_(self, db):

# change the user and password as needed

connection = pymysql.connect(host = 'localhost',

user = 'root',

password = 'root',

db = db,

charset = 'utf8mb4',

cursorclass = pymysql.cursors.DictCursor,

autocommit = True)

# establish the connection to the database

self.connection = connection

# the method to query the database

def query\_db(self, query, data=None):

with self.connection.cursor() as cursor:

try:

query = cursor.mogrify(query, data)

print("Running Query:", query)

cursor.execute(query, data)

if query.lower().find("insert") >= 0:

# INSERT queries will return the ID NUMBER of the row inserted

self.connection.commit()

return cursor.lastrowid

elif query.lower().find("select") >= 0:

# SELECT queries will return the data from the database as a LIST OF DICTIONARIES

result = cursor.fetchall()

return result

else:

# UPDATE and DELETE queries will return nothing

self.connection.commit()

except Exception as e:

# if the query fails the method will return FALSE

print("Something went wrong", e)

return False

finally:

# close the connection

self.connection.close()

# connectToMySQL receives the database we're using and uses it to create an instance of MySQLConnection

def connectToMySQL(db):

return MySQLConnection(db)

\*\*\*\*\*\*\*\*\*\*\*\* flask\_app/controllers/user\_controll.py \*\*\*\*\*\*\*\*\*\*

from flask import render\_template, session, redirect, request,flash

from flask\_app import app

from flask\_app.models.users\_model import User

from flask\_bcrypt import Bcrypt

bcrypt = Bcrypt(app)

@app.route('/')

def main():

return render\_template("index.html")

@app.route('/new',methods=['post'])

def add\_new():

if User.validation(request.form):

hashed\_password=bcrypt.generate\_password\_hash(request.form['password'])

print(request.form)

user\_data={

\*\*request.form,

'password':hashed\_password

}

print(request.form['password'],"="\*20,hashed\_password)

session['user\_id']=User.add\_user(user\_data)

return redirect('/')

return render\_template("index.html")

@app.route('/user/login', methods=['post'])

def login\_user():

data={

"email":request.form['email'],

"password":request.form['password']

}

user\_in\_db=User.get\_by\_email({'email': request.form['email']})

if user\_in\_db:

if not bcrypt.check\_password\_hash(user\_in\_db.password, data['password']):

flash('Invalid password/Email')

return redirect('/')

else:

session['user\_id']=user\_in\_db.id

return redirect('/dashboard')

flash('Invalid password/Email')

return redirect('/')

@app.route('/dashboard')

def dashboard():

user\_logged\_in=User.get\_by\_id({"id": session['user\_id']})

return render\_template("dashboard.html",loggedin=user\_logged\_in)

@app.route('/logout')

def logout\_user():

session.clear()

return redirect('/')

\*\*\*\*\*\*\*\*\*\* flask\_app/models/users\_model.py \*\*\*\*\*\*\*\*\*\*\*\*\*

from flask\_app.config.mysqlconnection import connectToMySQL

from flask\_app import DATABASE

from flask import flash

import re

EMAIL\_REGEX = re.compile(r'^[a-zA-Z0-9.+\_-]+@[a-zA-Z0-9.\_-]+\.[a-zA-Z]+$')

class User:

def \_\_init\_\_(self,data):

self.id=data['id']

self.first\_name=data['first\_name']

self.last\_name=data['last\_name']

self.email=data['email']

self.password=data['password']

self.created\_at=data['created\_at']

self.updated\_at=data['updated\_at']

@classmethod

def add\_user(cls,data):

query="""

INSERT INTO users (first\_name, last\_name, email, password)

VALUES (%(first\_name)s,%(last\_name)s,%(email)s,%(password)s);

"""

return connectToMySQL(DATABASE).query\_db(query,data)

@classmethod

def get\_by\_email(cls,data):

query=""" SELECT \* FROM users WHERE email = %(email)s;"""

results=connectToMySQL(DATABASE).query\_db(query,data)

print("="\*5,results,"="\*5)

if len(results)<1:

return False

return cls(results[0])

@classmethod

def get\_by\_id(cls,data):

query="""SELECT \* FROM users WHERE id = %(id)s;"""

results=connectToMySQL(DATABASE).query\_db(query,data)

print("="\*15,results,"="\*15)

if len(results)<1:

return False

return cls(results[0])

@staticmethod

def validation(data):

is\_valid = True

if len(data['first\_name'])<2:

is\_valid = False

flash("A Valid First Name is required !!!!!")

if len(data['last\_name'])<2:

is\_valid = False

flash("A Valid Last Name is required !!!")

if not EMAIL\_REGEX.match(data['email']):

is\_valid = False

flash("A valid email is required!!!!")

elif User.get\_by\_email({'email':data['email']}):

is\_valid = False

flash("User Already exist")

if len(data['password'])<7:

is\_valid = False

flash("you must respect password caracter number")

elif data['password'] != data['confirm\_password']:

is\_valid = False

flash("password and confirmation password must matchs")

return is\_valid

\*\*\*\*\*\*\*\*\*\* flask\_app/templates/index.html \*\*\*\*\*\*

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>Home</title>

</head>

<body>

<div class="container"style="justify-content: center;">

<div class="row" style="margin-top: 40px; display: flex; justify-content: space-around;">

<div class="col-4">

<h2>Register</h2>

<form action="/new" method="post">

{% for message in get\_flashed\_messages()%}

<p class="text-danger"> {{message}}</p>

{%endfor%}

<div class="form-group">

<label for="">First Name</label>

<input type="text" name="first\_name" class="form-control">

</div>

<div class="form-group">

<label for="">Last Name</label>

<input type="text" name="last\_name" class="form-control">

</div>

<div class="form-group">

<label for="">Email</label>

<input type="email" name="email" class="form-control">

</div>

<div class="form-group">

<label for="">Password</label>

<input type="password" name="password" class="form-control">

</div>

<div class="form-group">

<label for="">Confirm Password</label>

<input type="password" name="confirm\_password" class="form-control">

</div>

<button class="btn btn-info">Create</button>

</form>

</div>

<div class="col-4">

<h2>Login</h2>

<form action="/user/login" method="post">

{% for message in get\_flashed\_messages()%}

<p class="text-danger"> {{message}}</p>

{%endfor%}

<div class="form-group">

<label for="">Email</label>

<input type="text" name="email" class="form-control">

</div>

<div class="form-group">

<label for="">Password</label>

<input type="password" name="password" class="form-control">

</div>

<button class="btn btn-success">Login</button>

</form>

</div>

</div>

</div>

</body>

</html>

\*\*\*\*\*\*\* flask\_app/templates/dashboard.html \*\*\*\*\*\*\*\*\*\*

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"

integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<title>Dashboard</title>

</head>

<body>

<div class="container">

<h1 style="justify-items: center; margin-top: 20px; margin-left: 40px;"> Welcome {{loggedin.first\_name}}</h1>

<a href="/" class="btn btn-info" style="margin: 40px;">Logout</a>

</div>

</body>

</html>