Assignment 2

Date	20 October 2022
Name	A.Niranjana
RollNo	910619104053
Maximum Marks	2 Marks

Question 1 :

Create User table with email,username,roll number, password.

Solution :

```
create table userdetails(
uid integer unique not null GENERATED ALWAYS AS IDENTITY (START
WITH1 INCREMENT BY 1), rollno integer unique not null,
username varchar(30) unique not null,email
varchar(40) unique not null, password
varchar(8) unique not null
)
```

Question 2 :

Perform UPDATE,DELETE Queries with user table

Solution :

```
import ibm_db
def printTableData(conn):
    sql = "SELECT * FROM userdetails"
    out = ibm_db.exec_immediate(conn, sql)
    document = ibm_db.fetch_assoc(out) while
    document != False:

dictionary.update({document['USERNAME']:document['PA
```

```python
SSWORD']})                    document =
ibm_db.fetch_assoc(out)
        def insertTableData(conn,rollno,username,email,password):sql="INSERT
            INTO
userdetails(rollno,username,email,password) VALUES
({},'{}','{}','{}')".format(rollno,username,email,password)


        out = ibm_db.exec_immediate(conn,sql)
        print('Number of affected rows : ',ibm_db.num_rows(out),"\n")def
    updateTableData(conn,rollno,username,email,password):
        sql = "UPDATE userdetails SET
(username,email,password)=('{}','{}','{}') WHERE
rollno={}".format(username,email,password,rollno)
        out = ibm_db.exec_immediate(conn, sql)
        print('Number of affected rows : ', ibm_db.num_rows(out), "\n")def
    deleteTableData(conn,rollno):
        sql = "DELETE FROM userdetails WHERE rollno={}".format(rollno)
        out = ibm_db.exec_immediate(conn, sql)
        print('Number of affected rows : ', ibm_db.num_rows(out), "\n")
    try:
      conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=0c77d6f2-5da9-48a9-81f8-
86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31198;SECU
RITY=SSL;SSLServerCertifi
cate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=bjn03696;PWD=ef96tLJX2VjzaC
PX;", "", "")      print("Db connected")
    except:
      print("Error")

      from flask import Flask,render_template,request,url_for,sessionapp=Flask(_name_
      )
      @app.route("/")
```

```python
@app.route("/login",methods=['POST','GET']) def
login():
    if request.method=="POST":
        printTableData(conn)
        username=request.form['username']
        password=request.form['password'] try:
            if dictionary[username] == password  and username in dictionary:return
                "Logged in successfully"
        except:
                return "Invalid username or password" return
                render_template('loginpage.html')


  @app.route("/register",methods=['POST','GET']) def
 register():
        if request.method=="POST":
            rollno = request.form['rollno'] username =
            request.form['username']email =
            request.form['email'] password =
            request.form['password']
            insertTableData(conn, rollno, username, email, password)
           #printTableData(conn)
           return render_template('loginpage.html')
 return  render_template('registerpage.html')

#deleteTableData(conn,6)
#updateTableData(conn,5,'pradhee','pradhee@google.com','assignment') #printTableData(conn)
#print(dictionary)
if__name__=="_main_":app.run(debug=True)
```

Question 3 :

Connect python code to db2.

Question 4 :

Create a flask app with registration page, login page and welcome page. By default load the registration page   once the user enters all the fields store the data in database and navigateto login page.Authenticate user with   username and password. If the user is valid show the welcome page.

Forms.py

```python
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SubmitField, BooleanFieldfrom
wtforms.validators import DataRequired, Length, Email, EqualTo
class RegistrationForm(FlaskForm):
    username = StringField('Username',validators=[DataRequired(),
Length(min=2, max=20)])          email =
StringField('Email',validators=[DataRequired(), Email()])
    password = PasswordField('Password', validators=[DataRequired()]) confirm_password =
    PasswordField('Confirm Password', validators=[DataRequired(),
EqualTo('password')])          submit = SubmitField('Sign Up')
class LoginForm(FlaskForm):
    email = StringField('Email',validators=[DataRequired(), Email()]) password =
    PasswordField('Password', validators=[DataRequired()])remember =
    BooleanField('Remember Me')
    submit = SubmitField('Login')
```

Layout.html

```html
<!DOCTYPE html>
<html lang="en">
<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Home Page</title>

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">

    {% if title %}
      <title>{{ title }}</title>
    {% else %}
      <title>Home page</title>
    {% endif %}

  </head>
 <body>
  <header class="site-header">
    <nav class="navbar navbar-expand-md navbar-dark bg-steel fixed-top">
     <div class="container">
       <a class="navbar-brand mr-4" href="/">Nutrition Assistant Application</a>
       <button class="navbar-toggler" type="button" data-toggle="collapse" data- target="#navbarToggle" aria
controls="navbarToggle" aria-expanded="false" aria-
```

```html
    label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarToggle">
        <div class="navbar-nav mr-auto">
          <a class="nav-item nav-link" href="/">Home</a>
          <a class="nav-item nav-link" href="/about">About</a>
        </div>
        <!-- Navbar Right Side -->
        <div class="navbar-nav">
          <a class="nav-item nav-link" href="/login">Login</a>
          <a class="nav-item nav-link" href="/register">Register</a>
        </div>
      </div>
     </div>
    </nav>
  </header>
  <main role="main" class="container">
   <div class="row">
     <div class="col-md-8">
       {% block content %} {% endblock %}
     </div>
     <div class="col-md-4">
       <div class="content-section">
         <p class='text-muted'>Here you Welcome! To our Nutrition Assistant We are
           here to help you to maintain your physic.</p>
       </div>
     </div>
   </div>
  </main>
  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
```

KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/
GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

```html
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
 </body>
 </html>
```

Login.html

```html
 {% extends "layout.html" %}
 {% block content %}
  <div class="content-section">
    <form method="POST" action="">
      {{ form.hidden_tag() }}
      <fieldset class="form-group">
        <legend class="border-bottom mb-4">Log In</legend>
        <div class="form-group">
          {{ form.email.label(class="form-control-label") }}
          {% if form.email.errors %}
            {{ form.email(class="form-control form-control-lg is-invalid") }}
            <div class="invalid-feedback">
              {% for error in form.email.errors %}
                <span>{{ error }}</span>
              {% endfor %}
            </div>
          {% else %}
            {{ form.email(class="form-control form-control-lg") }}
```

```html
                {% endif %}
        </div>
        <div class="form-group">
            {{ form.password.label(class="form-control-label") }}
            {% if form.password.errors %}
                {{ form.password(class="form-control form-
control-lg is-invalid") }}                 <div class="invalid-
feedback">
                    {% for error in form.password.errors %}
                      <span>{{ error }}</span>
                    {% endfor %}
                </div>
            {% else %}
                {{ form.password(class="form-
control form-control-lg") }}                 {% endif
%}
        </div>
        <div class="form-check">
            {{ form.remember(class="form-check-input") }}
            {{ form.remember.label(class="form-check-label") }}
        </div>
      </fieldset>
      <div class="form-group">
          {{ form.submit(class="btn btn-outline-info") }}
      </div>
      <small class="text-muted ml-2">
        <a href="#">Forgot Password?</a>
      </small>
    </form>
  </div>
  <div class="border-top pt-3">
    <small class="text-muted">
      Need An Account? <a class="ml-2" href="{{ url_for('register')
```

```
}}">Sign Up Now</a>        </small>
   </div>
 {% endblock content %}
    Main.py

    from flask import Flask,
render_template, url_for, flash, redirect
from Forms import RegistrationForm,
LoginForm
    app = Flask(_name_)
    app.config['SECRET_KEY'] =
'5791628bb0b13ce0c676dfde280ba245' posts =
[
   {
     'title': 'Blog post I',
     'Author': 'Agenis',
     'date_posted': 'sep 25,2022','content':
     '1st post'
   },

   {
     'title': 'Blog post II',
     'Author': ' Aved ',
     'date_posted': 'sep 26,2022','content':
     '2st post'
   }
]

 @app.route("/")
 def home():
   return render_template('home.html', title='Home')

 @app.route("/about")
 def about():
```

```python
    return render_template('about.html', title='about')


@app.route("/register", methods=['GET', 'POST'])def
register():
    form = RegistrationForm()
    if form.validate_on_submit():
        flash(f'Account created for
{form.username.data}!', 'success') return
redirect(url_for('home'))
    return render_template('register.html', title='Register', form=form)@app.route("/login",
methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        if form.email.data == 'admin@blog.com' and
form.password.data == 'password':                flash('You have been
logged in!', 'success')
            return redirect(url_for('home'))
        else:
            flash('Login Unsuccessful. Please check username and
password', 'danger')                return
render_template('login.html', title='Login', form=form)


if __name__ == '_main_':app.run(debug=True)
```

Register.html

```html
{% extends "layout.html" %}
{% block content %}
<div class="content-section">
    <form method="POST" action="">
        {{ form.hidden_tag() }}
```

```html
<fieldset class="form-group">
    <legend class="border-bottom mb-4">Join Today</legend>
        <div class="form-group">
            {{ form.username.label(class="form-control-label") }}

            {% if form.username.errors %}
                {{ form.username(class="form-control form-control-lg is-invalid") }}
                <div class="invalid-feedback">
                    {% for error in form.username.errors %}
                        <span>{{ error }}</span>
                    {% endfor %}
                </div>
            {% else %}
                {{ form.username(class="form-control form-control-lg") }}
            {% endif %}
        </div>
        <div class="form-group">
            {{ form.email.label(class="form-control-label") }}
            {% if form.email.errors %}
                {{ form.email(class="form-control form-control-lg is-invalid") }}
                <div class="invalid-feedback">
                    {% for error in form.email.errors %}
                        <span>{{ error }}</span>
                    {% endfor %}
                </div>
            {% else %}
                {{ form.email(class="form-control form-control-lg") }}
            {% endif %}
        </div>
```

```html
<div class="form-group">
    {{ form.password.label(class="form-control-label") }}
    {% if form.password.errors %}
        {{ form.password(class="form-control form-control-lg is-invalid") }}
        <div class="invalid-feedback">
            {% for error in form.password.errors %}
                <span>{{ error }}</span>
            {% endfor %}
        </div>
    {% else %}
        {{ form.password(class="form-control form-control-lg") }}
    {% endif %}
</div>
<div class="form-group">
    {{ form.confirm_password.label(class="form-control-label") }}
    {% if form.confirm_password.errors %}
        {{ form.confirm_password(class="form-control form-control-lg is-invalid") }}
        <div class="invalid-feedback">
            {% for error in form.confirm_password.errors %}
                <span>{{ error }}</span>
            {% endfor %}
        </div>
    {% else %}
        {{ form.confirm_password(class="form-control form-control-lg") }}
    {% endif %}
</div>
</fieldset>
<div class="form-group">
    {{ form.submit(class="btn btn-outline-info") }}
</div>
```

```
        </form>
    </div>
    <div class="border-top pt-3">
        <small class="text-muted">
            Already Have An Account? <a class="ml-2" href="{{
url_for('login') }}">Sign In</a>          </small>
    </div>
  {% endblock content %}
```