

Name	Jeet Golecha
Roll No	SSNCE195001043
Date	10 September 2022
Team ID	PNT2022TMID53061
Project Name	Project - Personal Expense Tracker

## Assignment - 2

### CRUD Operations Using Flask

1. Create a User table with users with email,username,roll number, password.
2. Perform UPDATE,DELETE Queries with user table
3. Connect python code to db2.
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 the database and navigate to the login page to authenticate user username and password. If the user is valid, show the welcome page.

#### Source Code:

##### app.py

```
from flask import Flask, render_template, url_for, redirect, flash
from flask_login import UserMixin, login_user, LoginManager,
login_required, logout_user, current_user
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from wtforms import StringField, PasswordField, SubmitField, IntegerField
from flask_wtf import FlaskForm
import sqlite3

app = Flask(__name__)
bcrypt = Bcrypt(app)
app.config['SECRET_KEY'] = 'B7-1A3E'

login_manager = LoginManager()
```

```
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
def load_user(user_id):
    conn = connect_db()
    user = conn.execute('SELECT * FROM user WHERE id = ?',
                        (user_id,)).fetchone()
    usr_obj = User(user[0], user[1], user[2])
    return usr_obj

def connect_db():
    conn = sqlite3.connect('database.db')
    return conn

class User:

    def __init__(self, id, email, username):
        self.id = id
        self.username = username
        self.email = email

    def to_json(self):
        return {"username": self.username,
                "email": self.email}

    def is_authenticated(self):
        return True

    def is_active(self):
        return True

    def is_anonymous(self):
        return False

    def get_id(self):
        return str(self.id)
```

```

class RegisterForm(FlaskForm):
    email = StringField(validators=[
        InputRequired(), Length(min=4, max=50)], render_kw={"placeholder":
"Email"})
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=20)],
render_kw={"placeholder": "Username"})
    rollnumber = StringField(validators=[
        InputRequired(), Length(min=5, max=10)], render_kw={"placeholder":
"RollNumber"})
    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)],
render_kw={"placeholder": "Password"})

    submit = SubmitField('Register')

    def validate_username(self, username):
        conn = connect_db()
        existing_user_username = conn.execute('SELECT * FROM user WHERE
username = ?',
                                                (username.data,)).fetchone()

        conn.commit()
        conn.close()
        if existing_user_username:
            raise ValidationError(
                'That username already exists. Try another one.')

class LoginForm(FlaskForm):
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=20)],
render_kw={"placeholder": "Username"})

    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)],
render_kw={"placeholder": "Password"})

    submit = SubmitField('Login')

```

```

class UpdateForm(FlaskForm):
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=20)],
        render_kw={"placeholder": "Username"})

    oldpassword = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Previous Password"})

    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)],
        render_kw={"placeholder": "Password"})

    submit = SubmitField('Update')

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        conn = connect_db()
        user = conn.execute('SELECT * FROM user WHERE username = ?',
                             (form.username.data,)).fetchone()

        conn.commit()
        conn.close()
        if user:
            if bcrypt.check_password_hash(user[4], form.password.data):
                usr_obj = User(user[0], user[1], user[2])
                login_user(usr_obj)
                return redirect(url_for('welcome'))

            else:
                print('Hi')
                flash(f'Invalid credentials, check and try logging in
again.', 'danger')

```

```

        return redirect(url_for('login'))

    return render_template('login.html', form=form)

@app.route('/welcome', methods=['GET', 'POST'])
@login_required
def welcome():
    return render_template('welcome.html')

@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('login'))

@app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()
    conn = connect_db()
    if form.validate_on_submit():
        hashed_password =
bcrypt.generate_password_hash(form.password.data)
        conn.execute('INSERT INTO user (email, username, roll_number,
pass_word) VALUES (?, ?, ?, ?)',
                    (form.email.data, form.username.data,
form.rollnumber.data, hashed_password))
        conn.commit()
        conn.close()
        return redirect(url_for('login'))
    return render_template('register.html', form=form)

@app.route('/update', methods=['GET', 'POST'])
def update():
    form = UpdateForm()
    conn = connect_db()
    if form.validate_on_submit():

```

```

conn = connect_db()
user = conn.execute('SELECT * FROM user WHERE username = ?',
                    (form.username.data,)).fetchone()

if user:
    if bcrypt.check_password_hash(user[4], form.oldpassword.data):
        print(user)
        hashed_password1 = bcrypt.generate_password_hash(
            form.password.data)
        conn.execute('UPDATE user set pass_word = ? where username
= ?',
                    (hashed_password1, form.username.data))
        conn.commit()
        conn.close()
        flash(f'Password changed successfully.', 'success')
        return redirect(url_for('home'))
    else:
        flash(f'Invalid password, Enter valid password.',
'danger')
        return redirect(url_for('update'))
else:
    flash(f'Invalid user, Enter valid User.', 'danger')
    return redirect(url_for('update'))
return render_template('update.html', form=form)

if __name__ == "__main__":
    app.run(debug=True)

```

## schema.sql

```

DROP TABLE IF EXISTS user;

CREATE TABLE user (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    email TEXT NOT NULL,
    username TEXT NOT NULL,
    roll_number INTEGER NOT NULL,
    pass_word TEXT NOT NULL
);

```

## Init\_db.py

```
import sqlite3

conn = sqlite3.connect('database.db')

with open('schema.sql') as f:
    conn.executescript(f.read())

conn.commit()
conn.close()
```

## Style.css

```
.header {
    padding: 5px 120px;
    width: 150px;
    height: 70px;
    background-color: #236b8e;
}

.border {
    padding: 80px 50px;
    width: 400px;
    height: 450px;
    border: 1px solid #236b8e;
    border-radius: 0px;
    background-color: #9ac0cd;
}

.btn {
    padding: 10px 40px;
    background-color: #236b8e;
    color: #ffffff;
    font-style: oblique;
    font-weight: bold;
    border-radius: 10px;
}
```

```
.textbox {
  padding: 10px 40px;
  background-color: #236b8e;
  text-shadow: #ffffff;
  border-radius: 10px;
}

::placeholder {
  color: #ffffff;
  opacity: 1;
  font-style: oblique;
  font-weight: bold;
}

.word {
  color: #ffffff;
  font-style: oblique;
  font-weight: bold;
}

.bottom {
  color: #236b8e;
  font-style: oblique;
  font-weight: bold;
}

body {
  background: #fafafa;
  color: #333333;
  margin-top: 5rem;
}

h1, h2, h3, h4, h5, h6 {
  color: #444444;
}

.bg-steel {
  background-color: #5f788a;
}
```



```
.site-header .navbar-nav .nav-link {
  color: #cbd5db;
}

.site-header .navbar-nav .nav-link:hover {
  color: #ffffff;
}

.site-header .navbar-nav .nav-link.active {
  font-weight: 500;
}

.content-section {
  background: #ffffff;
  padding: 10px 20px;
  border: 1px solid #dddddd;
  border-radius: 3px;
  margin-bottom: 20px;
}

.article-title {
  color: #444444;
}

a.article-title:hover {
  color: #428bca;
  text-decoration: none;
}

.article-content {
  white-space: pre-line;
}

.article-img {
  height: 65px;
  width: 65px;
  margin-right: 16px;
}

.article-metadata {
```

```
padding-bottom: 1px;
margin-bottom: 4px;
border-bottom: 1px solid #e3e3e3
}

.article-metadata a:hover {
  color: #333;
  text-decoration: none;
}

.article-svg {
  width: 25px;
  height: 25px;
  vertical-align: middle;
}

.account-img {
  height: 125px;
  width: 125px;
  margin-right: 20px;
  margin-bottom: 16px;
}

.account-heading {
  font-size: 2.5rem;
}
```

## Templates:

### Welcome.html

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

<head>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Welcome Page</title>
</head>

<body>
  <h1>Welcome!</h1> <br>
  <a href="{{url_for('logout')}}">Press here to logout</a>

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDs4f4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>

</html>
```

## Login.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">
  <title>Login</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
</head>

<body>
  <div class="col-md-8">
    {% with messages = get_flashed_messages(with_categories=true) %}
      {% if messages %}
        {% for category, message in messages %}
          <div class="alert alert-{{category}}">
            {{ message }}
          </div>
        {% endfor %}
      {% endif %}
    {% endwith %}
    {% block content %} {% endblock %}
  </div>

  <h1>Login Page</h1> <br>

  <form method="POST" action="">
    {{ form.hidden_tag() }}
    <fieldset class="form-group">
      <div>
        {{ 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>
        {{ 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>

</fieldset>
<div class="form-group">
    {{ form.submit(class="btn btn-outline-info") }}
</div>

<small class="text-muted ml-2">

```

```

        <a href='{{url_for('register')}}'>Do not have an account? Sign
Up?</a>

    </small>

</form>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>

</html>

```

## Home.html

```

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />

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

```

```

</head>

<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{category}}">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        {% block content %} {% endblock %}
    </div>

    <h1>Flask Application</h1> <br>
    <a href="{{ url_for('login') }}">Login Page</a><br> <br>
    <a href="{{ url_for('register') }}">Register Page</a><br> <br>
    <a href="{{ url_for('update') }}">Update Password</a><br> <br>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>

    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDs4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>

</html>

```

## Register.html

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

<head>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Register</title>
</head>

<body>
  <h1>Register Page</h1> <br>

  <!-- <form method="POST" action="">
    {{ form.hidden_tag() }}
    {{ form.email }} <br> <br>
    {{ form.username }} <br> <br>
    {{ form.rollnumber }} <br> <br>
    {{ form.password }} <br> <br>
    {{ form.submit }} <br> <br>
  </form> -->

  <form method="POST" action="">
    {{ form.hidden_tag() }}
    <fieldset class="form-group">
      <!-- <legend class="border-bottom mb-4">Registration
Page</legend> -->

      <div>
        {{ 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>

<div>
    {{ 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>
    {{ form.rollnumber.label(class="form-control-label") }}

    {% if form.rollnumber.errors %}
        {{ form.rollnumber(class="form-control form-control-lg
is-invalid") }}

        <div class="invalid-feedback">
            {% for error in form.rollnumber.errors %}
                <span>{{ error }}</span>
            {% endfor %}
        </div>
    {% else %}
        {{ form.rollnumber(class="form-control form-control-lg")
}}
    {% endif %}
</div>

```

```

        {% endfor %}
    </div>
    {% else %}
        {{ form.rollnumber(class="form-control
form-control-lg") }}
    {% endif %}
</div>

<div>
    {{ 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>

</fieldset>
<div class="form-group">
    {{ form.submit(class="btn btn-outline-info") }}
</div>

<small class="text-muted ml-2">
    <a href="{{ url_for('login') }}">Already have an account? Log
In</a>
</small>

</form>

```

```

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js"
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/njGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>

</html>

```

## Update.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">
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
</head>

<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}

```

```
<div class="alert alert-{{category}}">
    {{ message }}
</div>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %} {% endblock %}

</div>

<h1>Update Password</h1> <br>

<form method="POST" action="">
    {{ form.hidden_tag() }}
    <fieldset class="form-group">
        <div>
            {{ 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>
            {{ form.oldpassword.label(class="form-control-label") }}

            {% if form.oldpassword.errors %}
                {{ form.oldpassword(class="form-control
form-control-lg is-invalid") }}

                <div class="invalid-feedback">
                    {% for error in form.oldpassword.errors %}
                        <span>{{ error }}</span>
                    {% endfor %}
                </div>
            {% else %}
                {{ form.oldpassword(class="form-control form-control-lg")
}}
            {% endif %}
        </div>
    </fieldset>
</form>
```

```

        {% endfor %}

    </div>

    {% else %}

        {{ form.oldpassword(class="form-control
form-control-lg") }}

    {% endif %}

</div>

<div>

    {{ 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>

</fieldset>

<div class="form-group">

    {{ form.submit(class="btn btn-outline-info") }}

</div>

</form>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>

<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdsSJKQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>

```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>

</html>
```

## Output Snapshots:

# Flask Application

[Login Page](#)

[Register Page](#)

[Update Password](#)

## Registration:

← → ↻ 127.0.0.1:5000/register 🔑 📄 ☆ ⚙️ 🗄️ 👤 ⋮

### Register Page

Email

Username

Rollnumber

Password

**Register**

[Already have an account? Log In](#)

## Login:

← → ↻ 127.0.0.1:5000/login 🔑 📄 ☆ ⚙️ 🗄️ 👤 ⋮

### Login Page

Username

Password

**Login**

[Do not have an account? Sign Up?](#)

← → ↻ 127.0.0.1:5000/login 🔑 📄 ☆ ⚙️ 🗄️ 👤 ⋮

Invalid credentials, check and try logging in again.

## Login Page

Username

Password

Login

[Do not have an account? Sign Up?](#)

← → ↻ 127.0.0.1:5000/welcome 🔑 📄 ☆ ⚙️ 🗄️ 👤 ⋮

## Welcome!

[Press here to logout](#)

## Password Updation:

← → ↻ 127.0.0.1:5000/update 🔑 📄 ☆ ⚙️ 🗄️ 👤 ⋮

## Update Password

Username

Oldpassword

Password

Update



←

→

↺

127.0.0.1:5000/update

🔑

🔗

☆

⚙️

🖱️

🟢

⋮

Invalid user, Enter valid User.

# Update Password

Username

Username

Oldpassword

Previous Password

Password

Password

Update

←

→

↺

127.0.0.1:5000/update

🔑

🔗

☆

⚙️

🖱️

🟢

⋮

# Update Password

Username

jeet123

Oldpassword

.....

Password

.....

Update

← → ↺ 127.0.0.1:5000/update 🔑 📄 ☆ ⚙️ 🖨️ 👤 ⋮

Invalid password, Enter valid password.

## Update Password

Username

*Username*

Oldpassword

*Previous Password*

Password

*Password*

Update

← → ↺ 127.0.0.1:5000/update 🔑 📄 ☆ ⚙️ 🖨️ 🌐 ⋮

Invalid password, Enter valid password.

Update Password

Username

jeet123

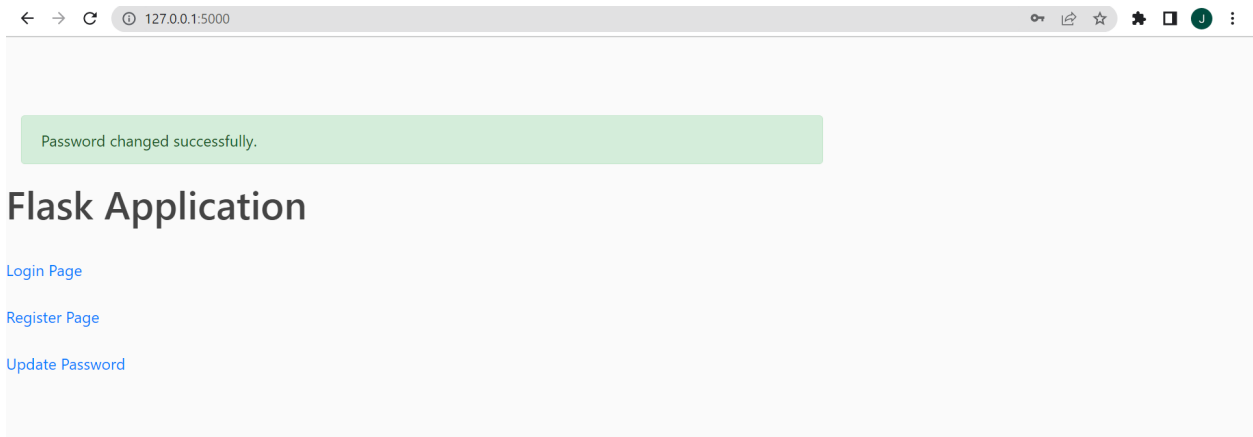
Oldpassword

.....

Password

.....

Update



## Login with new password:

