

# Login and Registration Project Using Flask and MySQL

Last Updated : 05 Sep, 2020

**Project Title:** Login and registration Project using Flask framework and MySQL Workbench.

**Type of Application (Category):** Web application.

**Introduction:** A framework is a code library that makes a developer's life easier when building web applications by providing reusable code for common operations. There are a number of frameworks for Python, including Flask, Tornado, Pyramid, and Django. Flask is a lightweight web application framework. It is classified as a micro-framework because it does not require particular tools or libraries.

**Pre-requisite:** Knowledge of Python, MySQL Workbench and basics of Flask Framework. Python and MySQL Workbench should be installed in the system. Visual studio code or Spyder or any code editor to work on the application.

**Technologies used in the project:** Flask framework, MySQL Workbench.

## Implementation of the Project:

### (1) Creating Environment

**Step-1:** Create an environment. Create a project folder and a venv folder within.

```
py -3 -m venv venv
```

**Step-2:** Activate the environment.

```
venv\Scripts\activate
```

**Step-3:** Install Flask.

```
pip install Flask
```

### (2) MySQL Workbench

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

Know more about it : <https://www.mysql.com/products/workbench/>

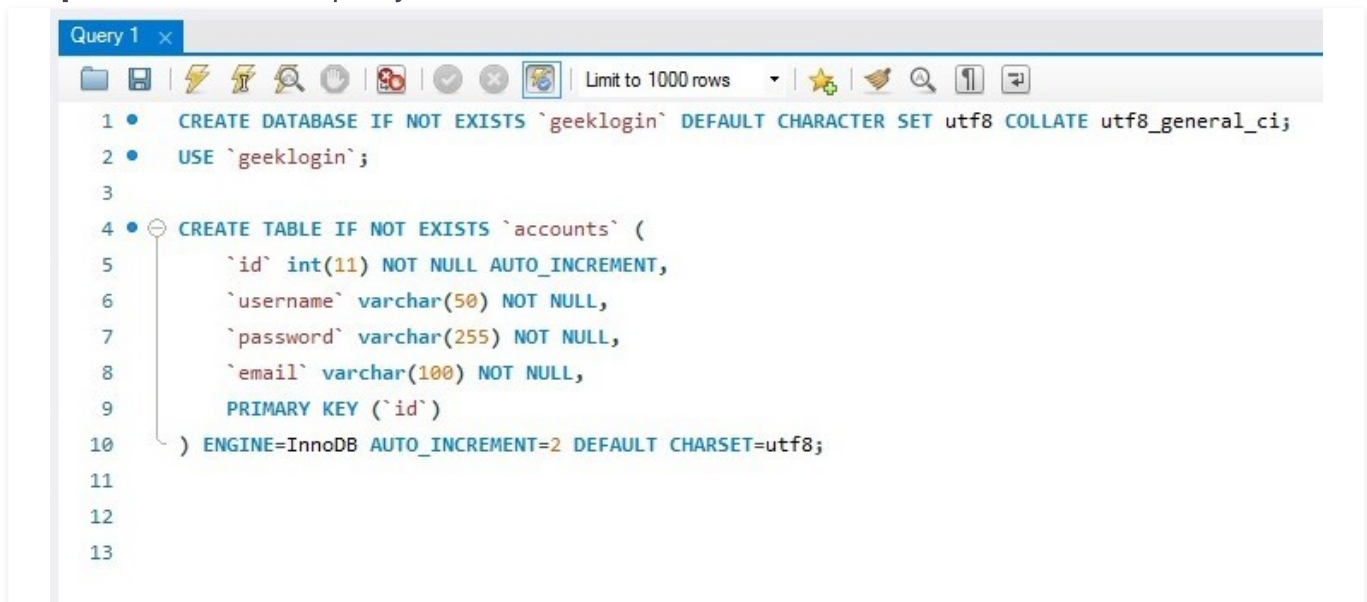
**Step-2:** Install 'mysqlbd' module in your venv.

```
pip install flask-mysqldb
```

**Step-3:** Open MySQL workbench.

**Step-4:** Write the following code. The above SQL statement will create our database **geeklogin** with the table **accounts**.

**Step-5:** Execute the query.



```
Query 1 x
1 • CREATE DATABASE IF NOT EXISTS `geeklogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
2 • USE `geeklogin`;
3
4 • CREATE TABLE IF NOT EXISTS `accounts` (
5     `id` int(11) NOT NULL AUTO_INCREMENT,
6     `username` varchar(50) NOT NULL,
7     `password` varchar(255) NOT NULL,
8     `email` varchar(100) NOT NULL,
9     PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
11
12
13
```

### (3) Creating Project

**Step-1:** Create an empty folder 'login'.

**Step-2:** Now open your code editor and open this 'login' folder.

**Step-3:** Create 'app.py' folder and write the code given below.

```
# Store this code in 'app.py' file
```

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysql import MySQL
import MySQLdb.cursors
import re
```

```
app = Flask(__name__)
```

```
app.secret_key = 'your secret key'
```

```
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'your password'
app.config['MYSQL_DB'] = 'geeklogin'
```

```
mysql = MySQL(app)
```

```
@app.route('/')
@app.route('/login', methods=['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM accounts WHERE username = %s AND password = %s')
        account = cursor.fetchone()
        if account:
            session['loggedin'] = True
            session['id'] = account['id']
            session['username'] = account['username']
            msg = 'Logged in successfully !'
            return render_template('index.html', msg=msg)
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg=msg)
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```

session.pop('id', None)
session.pop('username', None)
return redirect(url_for('login'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        email = request.form['email']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM accounts WHERE username = % s', (username,))
        account = cursor.fetchone()
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not password or not email:
            msg = 'Please fill out the form !'
        else:
            cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username, password, email))
            mysql.connection.commit()
            msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)

```

**Step-4:** Create the folder **'templates'**. create the file 'login.html', 'register.html', 'index.html' inside the 'templates' folder.

**Step-5:** Open **'login.html'** file and write the code given below. In 'login.html', we have two fields i.e. username and password. When user enters correct username and password, it will route you to index page otherwise 'Incorrect username/password' is displayed.

```

<!-- Store this code in 'login.html' file inside the 'templates' folder -->

<html>
<head>
    <meta charset="UTF-8">

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#)

**Got It !**

```

<div align="center">
  <div align="center" class="border">
    <div class="header">
      <h1 class="word">Login</h1>
    </div></br></br></br>
    <h2 class="word">
      <form action="{{ url_for('login') }}" method="post">
        <div class="msg">{{ msg }}</div>
        <input id="username" name="username" type="text" placeh
        <input id="password" name="password" type="password" pl
        <input type="submit" class="btn" value="Sign In"></br><
      </form>
    </h2>
    <p class="bottom">Dont't have an account? <a class="bottom" hr
  </div>
</div>
</body>
</html>

```

**Step-6:** Open 'register.html' file and write the code given below. In 'register.html', we have three fields i.e. username, password and email. When user enters all the information, it stored the data in the database and 'Registration successful' is displayed.

*<!-- Store this code in 'register.html' file inside the 'templates' folder .*

```

<html>
  <head>
    <meta charset="UTF-8">
    <title> Register </title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.c
  </head>
  <body></br></br></br></br></br>
    <div align="center">
      <div align="center" class="border">
        <div class="header">
          <h1 class="word">Register</h1>
        </div></br></br></br>
        <h2 class="word">
          <form action="{{ url_for('register') }}" method="post">
            <div class="msg">{{ msg }}</div>
            <input id="username" name="username" type="text" placeh
            <input id="password" name="password" type="password" pl
            <input id="email" name="email" type="text" placeholder=
            <input type="submit" class="btn" value="Sign In"></br>

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```

        </div>
    </div>
</body>
</html>

```

**Step-7:** Open '**index.html**' file and write the code given below. This page is displayed when login is successful and username is also displayed. The logout functionality is also included in this page. When user logs out, it moves to fresh login page again.

```

<!-- Store this code in 'index.html' file inside the 'templates' folder-->

<html>
  <head>
    <meta charset="UTF-8">
    <title> Index </title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.c
  </head>
  <body><br><br><br><br><br>
    <div align="center">
      <div align="center" class="border">
        <div class="header">
          <h1 class="word">Index</h1>
        </div><br><br><br>
        <h1 class="bottom">
          Hi {{session.username}}!!<br><br> Welcome to the inde
        </h1><br><br><br>
        <a href="{{ url_for('logout') }}" class="btn">Logout</a>
      </div>
    </div>
  </body>
</html>

```

**Step-8:** Create the folder '**static**'. create the file '**style.css**' inside the 'static' folder and paste the given CSS code.

```

/* Store this code in 'style.css' file inside the 'static' folder*/

.header{
  padding: 5px 120px;
  width: 150px;
  height: 70px;
}

```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

```
.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-color: #FFFFFF;
  border-radius: 10px;
}

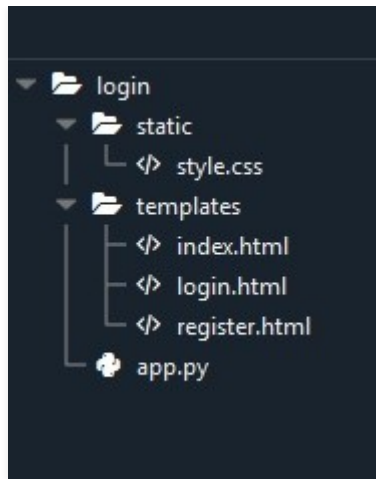
::placeholder {
  color: #FFFFFF;
  opacity: 1;
  font-style: oblique;
  font-weight: bold;
}

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

.bottom{
  color: #236B8E;
  font-style: oblique;
  font-weight: bold;
}
```

**Step-9:** The project structure will look like this.





#### (4) Run the Project

**Step-1:** Run the server.

**Step-2:** Browse the URL 'localhost:5000'.

**Step-3:** The output web page will be displayed.

#### (5) Testing of the Application

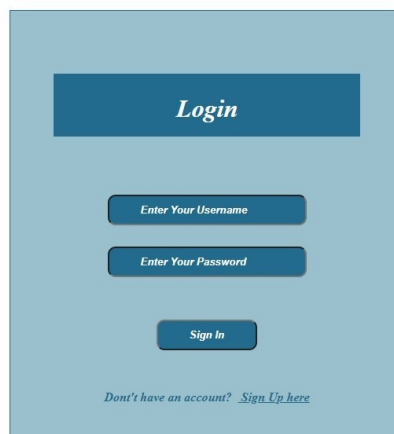
**Step-1:** If you are new user, go to sign up page and fill the details.

**Step-2:** After registration, go to login page. Enter your username and password and sign in.

**Step-3:** If your login is successful, you will be moved to index page and your name will be displayed.

**Output:**

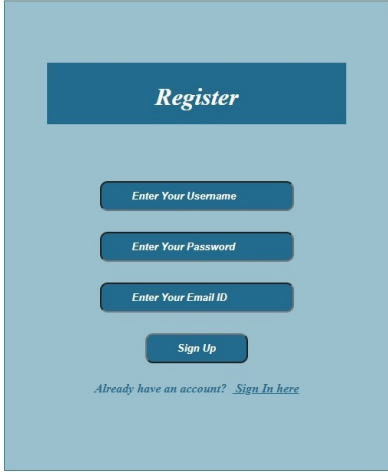
**Login page:**



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

## Registration page:



A screenshot of a registration form titled "Register". The form is set against a light blue background. It contains four input fields with blue borders and rounded corners, each with a placeholder text: "Enter Your Username", "Enter Your Password", "Enter Your Email ID", and "Sign Up". Below the "Sign Up" button, there is a link that says "Already have an account? [Sign In here](#)".

## If registration successful:

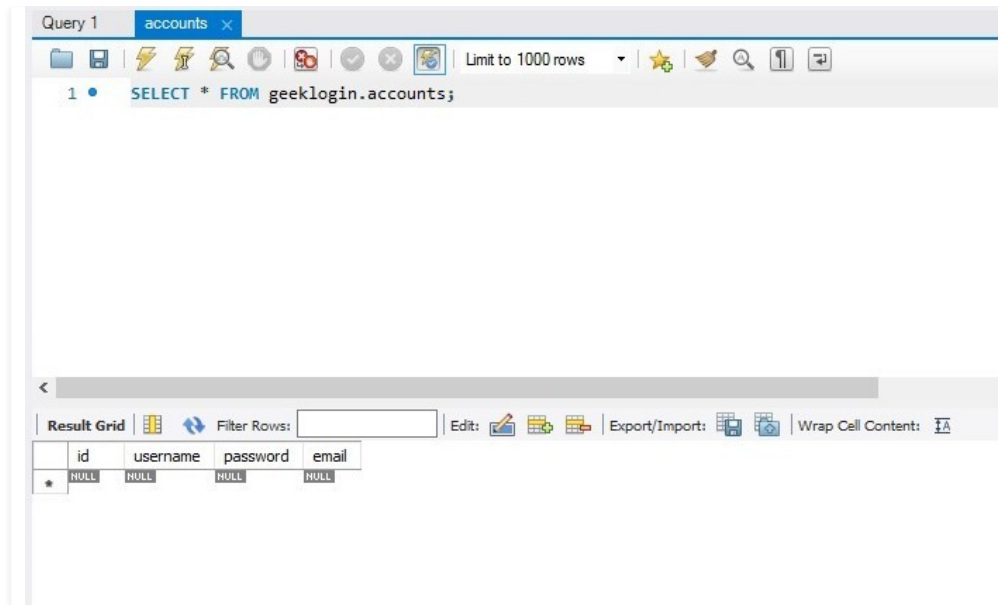


A screenshot of the same registration form after a successful registration. The form is titled "Register". Below the title, there is a message that says "You have successfully registered!". Below this message, there are four input fields with blue borders and rounded corners, each with a placeholder text: "Enter Your Username", "Enter Your Password", "Enter Your Email ID", and "Sign Up". Below the "Sign Up" button, there is a link that says "Already have an account? [Sign In here](#)".

## Before registration, Database table:

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**



Query 1 accounts

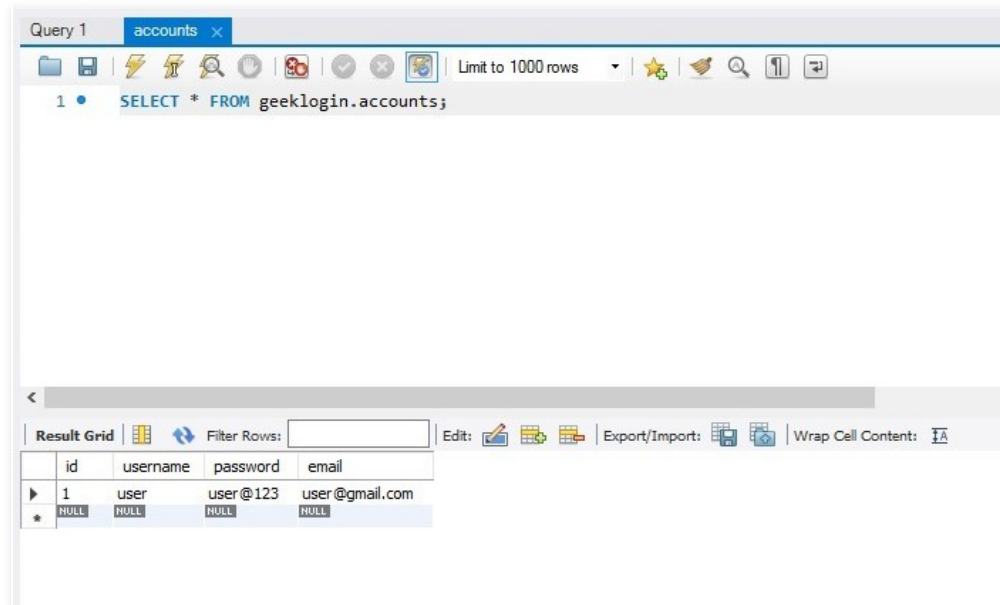
1 • `SELECT * FROM geeklogin.accounts;`

Limit to 1000 rows

Result Grid

	id	username	password	email
*	NULL	NULL	NULL	NULL

**After registration, Database table:**



Query 1 accounts

1 • `SELECT * FROM geeklogin.accounts;`

Limit to 1000 rows

Result Grid

	id	username	password	email
▶	1	user	user@123	user@gmail.com
*	NULL	NULL	NULL	NULL

**If login successful, Indexpage is displayed:**



## Related Articles

*Hi user!!*

*Welcome to the index page...*

[Logout](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

## If Login fails:



Attention geek! Strengthen your foundations with the [Python Programming Foundation](#) Course and learn the basics.

To begin with, your interview preparations Enhance your Data Structures concepts with the [Python DS](#) Course.

## RECOMMENDED ARTICLES

Page : **1** 2 3

**01** Profile Application using Python Flask and MySQL  
19, Jul 20

**05** Simple registration form using Python Tkinter  
13, Jun 18

**02** Flask project - Create a Joke App with PyJokes  
23, Nov 20

**06** How to get client\_id and client\_secret for Python Reddit API registration ?  
20, May 20

**03** Create MySQL Database Login Page in Python using Tkinter  
11, May 20

**07** Setup API for GeeksforGeeks user data using WebScraping and Flask  
23, Apr 19

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

**Like** 0[< Previous](#)[Next >](#)

### Article Contributed By :

**venniladeenan**

@venniladeenan

### Vote for difficulty

Easy

Normal

Medium

Hard

Expert

**Article Tags :** [Python Flask](#) , [Python](#) , [Web Technologies](#)[Improve Article](#)[Report Issue](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**



5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

[About Us](#)  
[Careers](#)  
[Privacy Policy](#)  
[Contact Us](#)

## Practice

[Courses](#)  
[Company-wise](#)  
[Topic-wise](#)  
[How to begin?](#)

## Learn

[Algorithms](#)  
[Data Structures](#)  
[Languages](#)  
[CS Subjects](#)  
[Video Tutorials](#)

## Contribute

[Write an Article](#)  
[Write Interview Experience](#)  
[Internships](#)  
[Videos](#)

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**