Build Blog website using Flask

Last Updated : 08 Mar, 2024

In this article, we'll explore how to build a dynamic blog website using Flask, a lightweight and versatile Python web framework. Flask provides developers with the tools needed to create robust web applications, and its simplicity makes it an excellent choice for beginners and experienced developers alike.

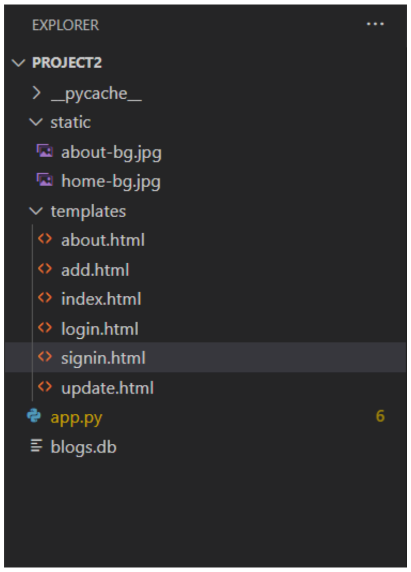
Requirements

- Basics of HTML & CSS
- Bootstrap
- Basic of Python
- Flask
- Flask-SQLAlchemy

Packages Required

```
pip install flask
pip install Flask-SQLAlchemy
```

File Structure



```
EXPLORER
PROJECT2
  __pycache__
  static
    about-bg.jpg
    home-bg.jpg
  templates
    about.html
    add.html
    index.html
    login.html
    signin.html
    update.html
  app.py
  blogs.db
```

Steps to Build Blog website using Flask

Step 1: Make a folder name as project2.

Step 2: We need to create 2 folders: "templates", and "statics".

Step 3: In templates create a file named an **index.html.**

HTML

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

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.
css" rel="stylesheet"
    integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

  <title>Home</title>
</head>

<body>

  <div id="part1">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Home</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarScroll"
          aria-controls="navbarScroll" aria-expanded="false" aria-
label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarScroll">
          <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll"
style="--bs-scroll-height: 100px;">
            <li class="nav-item">
              <a class="nav-link active" aria-current="page"
href="/">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link active" href="{{ url_for('about')
}}">About</a>
            </li>

            <li class="nav-item">
              <a class="nav-link active" href="{{ url_for('addpost') }}"
tabindex="-1">Add</a>
            </li>
          {% if name != "guest" %}
```

```
            <li class="nav-item">
              <a class="nav-link active" href="{{ url_for('signin') }}"
tabindex="-1">Signin</a>
            </li>
          {% endif %}

          </ul>
          <div class="mx-3">
            {% if name == "guest" %}
              <a class="btn btn-danger" href="{{ url_for('login')
}}">Login</a>

            {% else %}
              <a class="btn btn-danger" href="#">{{name}}</a>
              <a class="btn btn-danger" href="{{ url_for('logout')
}}">Logout</a>

            {% endif %}

          </div>
        </div>
      </div>
    </nav>
  </div>

  <div id="part2">
    <div id="carouselExampleCaptions" class="carousel slide" data-bs-
ride="carousel">
      <div class="carousel-indicators">
        <button type="button" data-bs-target="#carouselExampleCaptions"
data-bs-slide-to="0" class="active"
          aria-current="true" aria-label="Slide 1"></button>
        <button type="button" data-bs-target="#carouselExampleCaptions"
data-bs-slide-to="1"
          aria-label="Slide 2"></button>
      </div>
      <div class="carousel-inner">
        <div class="carousel-item active">
          <img src="../static/home-bg.jpg" class="d-block w-100"
alt="...">
          <div class="carousel-caption d-none d-md-block">
            <h5>First slide label</h5>
            <p>Some representative placeholder content for the first
slide.</p>
          </div>
        </div>
        <div class="carousel-item">
          <img src="../static/home-bg.jpg" class="d-block w-100"
alt="...">
          <div class="carousel-caption d-none d-md-block">
            <h5>Second slide label</h5>
            <p>Some representative placeholder content for the second
slide.</p>
          </div>
        </div>

      </div>

    </div>
  </div>

    <div id="part3">
      <div class="album py-5 bg-light">
        <div class="container">

          <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">


            <!-- <div class="col">
              <div class="card shadow-sm">
                <svg class="bd-placeholder-img card-img-top" width="100%"
height="225"
                  xmlns="http://www.w3.org/2000/svg" role="img" aria-
label="Placeholder: Thumbnail"
                  preserveAspectRatio="xMidYMid slice" focusable="false">
                  <title>Placeholder</title>
                  <rect width="100%" height="100%" fill="#55595c"></rect>
<text x="50%" y="50%" fill="#eceeef"
                    dy=".3em">Thumbnail</text>
                </svg>

                <div class="card-body">
                  <h2 class="post-title">
                    GFG Blog
                  </h2>
                  <p class="card-text">This is a wider card with
supporting text below as a natural lead-in to
                    additional content. This content is a little bit
longer.</p>
                  <div class="d-flex justify-content-between align-items-
center">
                    <div class="btn-group">
                      <button type="button" class="btn btn-sm btn-outline-
secondary">Edit</button>
                      <button type="button" class="btn btn-sm btn-outline-
secondary">Delete</button>
                    </div>
                    <small class="text-muted">9 mins</small>
                  </div>
                </div>
              </div>
            </div> -->
         <!-- loop to show all blogs that are present in the database -->
            {% for articles in article %}
            <div class="col">
              <div class="card shadow-sm">
                <svg class="bd-placeholder-img card-img-top" width="100%"
height="225"
                  xmlns="http://www.w3.org/2000/svg" role="img" aria-
label="Placeholder: Thumbnail"
                  preserveAspectRatio="xMidYMid slice" focusable="false">
                  <title>Placeholder</title>
                  <rect width="100%" height="100%" fill="#55595c"></rect>
<text x="50%" y="50%" fill="#eceeef"
                    dy=".3em">Thumbnail</text>
                </svg>

                <div class="card-body">
                  <h2 class="post-title">
                    {{ articles.title }}
                  </h2>
                  <p class="card-text">{{articles.content}}</p>
                  <p class="post-meta">Posted by {{ articles.author }}
</p>
                  <div class="d-flex justify-content-between align-items-
center">
                    <div class="btn-group">
                    <!-- Check if the user is not guest, if guest "edit
and delete button will be hidden" -->
                    {% if name != "guest" %}
                      <a href="/update/{{articles.id}}" type="button"
class="btn btn-sm btn-outline-secondary">Edit</a>
                      <a href="/delete/{{articles.id}}" type="button"
class="btn btn-sm btn-outline-secondary">Delete</a>
                    {% endif %}
                    </div>
                    <small class="text-muted">{{
articles.post_date.strftime('%B %d, %Y') }}</small>
                  </div>
                </div>
              </div>
            </div>
            {% endfor %}

          </div>
        </div>
      </div>
    </div>


  <div>

  </div>

  <!-- Optional JavaScript; choose one of the two! -->

  <!-- Option 1: Bootstrap Bundle with Popper -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle
.min.js"
    integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
    crossorigin="anonymous"></script>

</body>

</html>
```

Step 4: Create an **about.html** page.

HTML

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

```

```
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/boot
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65V

  <title>About</title>
</head>

<body>

  <div id="part1">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">GFG Blog</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="col
          aria-controls="navbarScroll" aria-expanded="false" aria-label=
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarScroll">
          <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll"
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="/">Ho
            </li>
            <li class="nav-item">
              <a class="nav-link" href="./about.html">About</a>
            </li>

            <li class="nav-item">
              <a class="nav-link" href="./add.html" tabindex="-1">Add</a
            </li>
          </ul>
          <form class="d-flex">
            <input class="form-control me-2" type="search" placeholder='
            <button class="btn btn-outline-success" type="submit">Search
          </form>
        </div>
      </div>
    </nav>
  </div>

  <div id="part2">
    <div id="carouselExampleCaptions" class="carousel slide" data-bs-ric
      <div class="carousel-indicators">
        <button type="button" data-bs-target="#carouselExampleCaptions"
          aria-current="true" aria-label="Slide 1"></button>
      </div>
      <div class="carousel-inner">
        <div class="carousel-item active">
          <img src="../static/about-bg.jpg" class="d-block w-100" alt=".
          <div class="carousel-caption d-none d-md-block">
            <h5>About Me</h5>
            <p>Some representative placeholder content for the first sli
          </div>
        </div>

      </div>

    </div>
  </div>

  <div class="container">
    <p class="lead ">Lorem ipsum, dolor sit amet consectetur adipisicing
      Ipsam ea, quam tempora quibusdam, ullam ex consequuntur illum
      dolor cum, aperiam illo obcaecati facere voluptatibus? Voluptas
      deserunt accusamus eius! Quo velit culpa obcaecati.</p>
  </div>

  <div>

  </div>

  <!-- Optional JavaScript; choose one of the two! -->

  <!-- Option 1: Bootstrap Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boot
    integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+
    crossorigin="anonymous"></script>

  <!-- Option 2: Separate Popper and Bootstrap JS -->
  <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bc
    -->
</body>

</html>
```

Step 5: Create an **add.html** page.

HTML

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

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"

  <title>Add Post</title>
</head>

```

```
<body>

  <div id="part1">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">GFG Blog</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
          aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigat
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarScroll">
          <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scr
            <li class="nav-item">
              <a class="nav-link active" aria-current="page" href="/">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="./about.html">About</a>
            </li>

            <li class="nav-item">
              <a class="nav-link" href="./add.html" tabindex="-1">Add</a>
            </li>
          </ul>
          <form class="d-flex">
            <input class="form-control me-2" type="search" placeholder="Search" aria-la
            <button class="btn btn-outline-success" type="submit">Search</button>
          </form>
        </div>
      </div>
    </nav>
  </div>

  <div id="part3">
    <div class="container">
      <div class="row">
        <div class="col-lg-8 col-md-10 mx-auto">
          <form name="addForm" id="addForm" method="POST" action="{{ url_for('addpost')
            <div class="control-group">
              <div class="form-group floating-label-form-group controls">
                <label>Title</label>
                <input type="text" class="form-control" placeholder="Title" name="title
                  data-validation-required-message="Please enter a title.">
                <p class="help-block text-danger"></p>
              </div>
            </div>

            <div class="control-group">
              <div class="form-group col-xs-12 floating-label-form-group controls">
                <label>Author</label>
                <input type="text" class="form-control" placeholder="Author" name="auth
                  data-validation-required-message="Please enter your phone number.">
                <p class="help-block text-danger"></p>
              </div>
            </div>
            <div class="control-group">
              <div class="form-group floating-label-form-group controls">
                <label>Blog Content</label>
```

```
                  <textarea rows="5" class="form-control" placeholder="Blog content" name
                    data-validation-required-message="Please enter a message."></textarea
                  <p class="help-block text-danger"></p>
                </div>
              </div>
              <div class="mb-3">
                <label for="formFileSm" class="form-label">Small file input example</labe
                <input class="form-control form-control-sm" id="formFileSm" type="file">
              </div>
              <br>
              <div id="success"></div>
              <div class="form-group">
                <button type="submit" class="btn btn-secondary" id="sendMessageButton">Se
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>

    <div>

    </div>

    <!-- Optional JavaScript; choose one of the two! -->

    <!-- Option 1: Bootstrap Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.mi
      integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
      crossorigin="anonymous"></script>

    <!-- Option 2: Separate Popper and Bootstrap JS -->
    <!--
      <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.
      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
      -->
  </body>

  </html>
```

Step 6: Create a **update.html** page.

Python3

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

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css
        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLA
```

```html

    <title>Update Post</title>
</head>

<body>

    <div id="part1">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <div class="container-fluid">
                <a class="navbar-brand" href="#">GFG Blog</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" 
                    aria-controls="navbarScroll" aria-expanded="false" aria-label="Togg
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarScroll">
                    <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="/">Hom
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="./about.html">About</a>
                        </li>

                        <li class="nav-item">
                            <a class="nav-link" href="./add.html" tabindex="-1">Add</a>
                        </li>
                    </ul>
                    <form class="d-flex">
                        <input class="form-control me-2" type="search" placeholder="Sea
                        <button class="btn btn-outline-success" type="submit">Search</b
                    </form>
                </div>
            </div>
        </nav>
    </div>

    <div id="part2">
        <div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel
            <div class="carousel-indicators">
                <button type="button" data-bs-target="#carouselExampleCaptions" data-bs
                    aria-current="true" aria-label="Slide 1"></button>
            </div>
            <div class="carousel-inner">
                <div class="carousel-item active">
                    <img src="../static/about-bg.jpg" class="d-block w-100" alt="...">
                    <div class="carousel-caption d-none d-md-block">
                        <h5>About Me</h5>
                        <p>Some representative placeholder content for the first slide.
                    </div>
                </div>

            </div>

        </div>
    </div>

    <div id="part3">
```

```
            <div class="container">
                <div class="row">
                    <div class="col-lg-8 col-md-10 mx-auto">
                        <form name="addForm" id="addForm" method="POST" action="/update/{{e
                            <div class="control-group">
                                <div class="form-group floating-label-form-group controls">
                                    <label>Update Title</label>
                                    <input type="text" class="form-control" value="{{edit.t
                                        required data-validation-required-message="Please e
                                    <p class="help-block text-danger"></p>
                                </div>
                            </div>

                            <div class="control-group">
                                <div class="form-group col-xs-12 floating-label-form-group 
                                    <label>Author</label>
                                    <input type="text" class="form-control" value="{{edit.a
                                        id="author" required
                                        data-validation-required-message="Please enter your
                                    <p class="help-block text-danger"></p>
                                </div>
                            </div>
                            <div class="control-group">
                                <div class="form-group floating-label-form-group controls">
                                    <label>Blog Content</label>


                                    <!-- <textarea rows="5" class="form-control"
                                        placeholder="{{edit.content}}" name="content" id="c
                                        data-validation-required-message="Please enter a me
                                        Geeksforgeeks
                                   </textarea> -->

                                    <input type="text" value="{{edit.content}}" name="conte
                                        class="form-control">
                                    <p class="help-block text-danger"></p>
                                </div>
                            </div>
                            <br>
                            <div id="success"></div>
                            <div class="form-group">
                                <button type="submit" class="btn btn-secondary" id="sendMes
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>

        <div>

        </div>

        <!-- Optional JavaScript; choose one of the two! -->

        <!-- Option 1: Bootstrap Bundle with Popper -->
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.
            integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIax
            crossorigin="anonymous"></script>

        <!-- Option 2: Separate Popper and Bootstrap JS -->
        <!--
        <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js"
        -->
    </body>

    </html>
```

Step 7: Create a **login.html** page.

HTML

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Login</title>
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}
```

```

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from
  the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}
</style>
</head>
<body>


  <form class="modal-content animate" action="{{ url_for('login') }}"
        method="POST">

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username"
             name="username" id="username" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password"
             name="password" id="password" required>

      <button type="submit">Login</button>
      <label>
        <input type="checkbox" checked="checked"
               name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button"
```

```
                onclick="document.getElementById('id01').style.display='none'"
                class="cancelbtn">Cancel</button>
        <span class="psw">Forgot <a href="#">password?</a></span>
      </div>
    </form>



</body>
</html>
```

Step 8: Create a **signin.html** page.

HTML

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Signup</title>

<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #04AA6D;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}

/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

```

```
/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15%
  from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
     display: block;
     float: none;
  }
  .cancelbtn {
     width: 100%;
  }
}
</style>
</head>
<body>


  <form class="modal-content animate"
        action="{{ url_for('signin') }}" method="POST">

    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username"
             name="username" id="username" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password"
             name="password" id="password" required>

      <button type="submit">Sign In</button>

    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button"
              onclick="document.getElementById('id01').style.display='none'"
              class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>

</body>
</html>
```

Step 9: Create an **app.py**
Import all libraries.
from distutils.log import debug
from flask import Flask, render_template, request, redirect, url_for
from flask_login import UserMixin, login_user, login_required, logout_user, current_user
from flask_sqlalchemy import SQLAlchemy
from flask_login import login_user, logout_user, login_required, LoginManager
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime
from flask import flash
Create a Flask application object and set the URI for the database to use.

```
app = Flask(__name__)
db = SQLAlchemy(app)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///blogs.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['SECRET_KEY'] = 'thisisasecretkey'
```

Create a class for our blog. Then use the application object as a parameter to create a GFGBLOG object of class SQLAlchemy.

```
class GFGBLOG(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(50))
    author = db.Column(db.String(20))
    post_date = db.Column(db.DateTime)
    content = db.Column(db.Text)
```

Create a class for users, and use the application object as a parameter to create a User object of class SQLAlchemy.

```
class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)
```

Here we will return the index.html whenever the route is called. The show all () function, which is tied to the '/' URL, is the application's entrance point. The HTML template receives the GFGBLOG table's record set as an input. The record is rendered as an HTML table by the template's server-side code.

```
@app.route("/")
def hello_world():
    article = GFGBLOG.query.order_by(GFGBLOG.post_date.desc()).all()
    print(current_user.is_anonymous)
    if current_user.is_anonymous:
        name = "guest"
    else:
        name = current_user.username
        print("bye")
    return render_template('index.html', article=article, name=name)
```

Here we will return the about.html whenever the **/about** is called.

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

Here we will return the about.html whenever the /about is called. If there is any POST request then the 'title', 'author', and 'content' will be saved in the database and it will redirect to the home page.

```
@app.route('/addpost', methods=['POST', 'GET'])
def addpost():
    if request.method == 'POST':
        title = request.form['title']
        author = request.form['author']
        content = request.form['content']
        post = GFGBLOG(title=title, author=author, content=content,
        post_date=datetime.now())
        db.session.add(post)
        db.session.commit()
        print("Done")
        return redirect(url_for('hello_world'))
    return render_template('add.html')
```

In the update, we will return the about.html whenever the /update is called. If there is any POST request for editing the blog then it will query the blog id from the database and then the 'title', 'author', and 'content' will be updated in the database and it will redirect to the home page.
Note: This can be only edited by a registered user, if a user is not registered then the edit button will be not shown to the Guest user.

```
@app.route('/update/<int:id>', methods=['POST', 'GET'])
@login_required
def update(id):
    if request.method == 'POST':
        title = request.form['title']
        author = request.form['author']
        content = request.form['content']
        print(content)
        post = GFGBLOG.query.filter_by(id=id).first()
        post.title = title
        post.author = author
        post.content = content
        db.session.add(post)
        db.session.commit()
        return redirect("/")
    edit = GFGBLOG.query.filter_by(id=id).first()
    return render_template('update.html', edit=edit)
```

In the delete, If there is any request for deleting the blog then it will query the blog id from the database then it will filter the first occurrence from the database and redirect to the home page.
Note: This can be only deleted by a registered user, if a user is not registered then the delete button will be not shown to the Guest user.

```
@app.route('/delete/<int:id>')
@login_required
def delete(id):
    d = GFGBLOG.query.filter_by(id=id).first()
    db.session.delete(d)
    db.session.commit()
    return redirect(url_for('hello_world'))
```

Here we will return the login.html whenever the login is called. If there is any POST request for logging a user then it will query the username from the database and if the user exits then it will log the user otherwise a message will flash to check user credentials.

```
@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        # print("hello")
        username = request.form['username']
        password = request.form['password']
        user = User.query.filter_by(username=username).first()
        if not user and not check_password_hash(user.password, password):
            flash('Please check your login details and try again.')
            # return redirect(url_for('auth.login'))
            return render_template('not.html')
        else:
            login_user(user)
            print("yes")
            return redirect(url_for('hello_world'))
    return render_template('login.html')
```

In the sign-in, we will return the signin.html whenever the **/signin** is called. If there is any POST request for registering a user for the blog then it will be saved in the User class database and it will redirect to the home page.

```
@app.route('/signin', methods=['POST', 'GET'])
def signin():
    if request.method == 'POST':
        print("hello")
        username = request.form['username']
        password = request.form['password']
        user = User(username=username, password=password)
        db.session.add(user)
        db.session.commit()
        return redirect(url_for('hello_world'))
    return render_template('signin.html')
```

Whenever a user clicks on a logout button then the app will log out that current user.

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

Code Implementation:

Python3

```python
from distutils.log import debug
from flask import Flask, render_template, request, redirect, url_for
from flask_login import UserMixin, login_user,
              login_required, logout_user, current_user
from flask_sqlalchemy import SQLAlchemy
from flask_login import login_user, logout_user, login_required, LoginManager
from werkzeug.security import generate_password_hash, check_password_hash
from datetime import datetime
from flask import flash


app = Flask(__name__)
db = SQLAlchemy(app)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///blogs.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['SECRET_KEY'] = 'thisisasecretkey'

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


@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))

class GFGBLOG(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(50))
    author = db.Column(db.String(20))
    post_date = db.Column(db.DateTime)
    content = db.Column(db.Text)

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20), nullable=False, unique=True)
    password = db.Column(db.String(80), nullable=False)



@app.route("/")
def hello_world():
    article = GFGBLOG.query.order_by(GFGBLOG.post_date.desc()).all()
    print(current_user.is_anonymous)
    if current_user.is_anonymous:
        name = "guest"
    else:
        name = current_user.username
        print("bye")

    return render_template('index.html', article=article, name=name)


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

```

```

@app.route('/addpost', methods=['POST', 'GET'])
def addpost():
    if request.method == 'POST':
        title = request.form['title']
        author = request.form['author']
        content = request.form['content']

        post = GFGBLOG(title=title, author=author,
                       content=content, post_date=datetime.now())

        db.session.add(post)
        db.session.commit()
        print("Done")
        return redirect(url_for('hello_world'))
    return render_template('add.html')

@app.route('/update/<int:id>', methods=['POST', 'GET'])
@login_required
def update(id):
    if request.method == 'POST':
        title = request.form['title']
        author = request.form['author']
        content = request.form['content']
        print(content)

        post = GFGBLOG.query.filter_by(id=id).first()

        post.title = title
        post.author = author
        post.content = content

        db.session.add(post)
        db.session.commit()
        return redirect("/")

    edit = GFGBLOG.query.filter_by(id=id).first()
    return render_template('update.html', edit=edit)

@app.route('/delete/<int:id>')
@login_required
def delete(id):
    d = GFGBLOG.query.filter_by(id=id).first()
    db.session.delete(d)
    db.session.commit()
    return redirect(url_for('hello_world'))

@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        # print("hello")
        username = request.form['username']
        password = request.form['password']
        user = User.query.filter_by(username=username).first()

        if not user and not check_password_hash(user.password, password):
            flash('Please check your login details and try again.')
            # return redirect(url_for('auth.login'))
            return render_template('not.html')
        else:
            login_user(user)
            print("yes")
            return redirect(url_for('hello_world'))

    return render_template('login.html')

@app.route('/signin', methods=['POST', 'GET'])
def signin():
    if request.method == 'POST':
        print("hello")
        username = request.form['username']
        password = request.form['password']

        user = User(username=username, password=password)
        db.session.add(user)
        db.session.commit()
        return redirect(url_for('hello_world'))

    return render_template('signin.html')


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

# main driver function
if __name__ == '__main__':

    # run() method of Flask class runs the application
    # on the local development server.
    app.run(debug=True)
```

Step 10: Create your database
Select cmd.
Type "**python**", to open the python console.
Type "**from <app_name> from db**".
Then type "**db.create_all()**", and you will see a database is created named **blogs.db** in the root directory.
Then you can type **exit()** to exit from the python console.

Output:
Note: For a first-time user you can register yourself by going directly to the link "**http://127.0.0.1:5000/signin**" which will show you to the sign-in page to register yourself in the database.
Looking to dive into the world of programming or sharpen your Python skills? Our **Master Python: Complete Beginner to Advanced Course** is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

R mailf... 1

Next Article
E-commerce Website using Django

Similar Reads

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses how endpoints ...

4 min read

Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would display minificatio...

12 min read

How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application.The Session is the time between the client logs in to the server and...

4 min read

How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is known as...

8 min read

Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used for building,...

11 min read

Create A Bookmark Organizer For Website Using Flask

This article explains how to make a tool called a Bookmark Organizer with Flask for a website. With this tool, we can add, create, update, delete, and edit bookmarks easily....

5 min read

Python | Build a REST API using Flask

Prerequisite: Introduction to Rest API REST stands for REpresentational State Transfer and is an architectural style used in modern web development. It defines a set or...

3 min read

Build a Text Translator Web App using Flask and Azure Cognitive Services

We're going to create a website that will translate text into multiple languages using Artificial Intelligence(AI). We'll use Flask framework for the Front end and Azure...

7 min read

How to Build a Web App using Flask and SQLite in Python

Python-based Flask is a microweb framework. Typically, a micro-framework has little to no dependencies on outside frameworks. Despite being a micro framework, practically...

4 min read

How to Build a Simple Android App with Flask Backend?

Flask is an API of Python that allows us to build up web applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is...

8 min read

Article Tags :
Geeks Premier League
Python
Geeks Premier League 2023
Python Flask

+1 More

Practice Tags :
python

Company

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation

OOAD	Puzzles
System Design Bootcamp	
Interview Questions	

School Subjects	GeeksforGeeks Videos
Mathematics	DSA
Physics	Python
Chemistry	Java
Biology	C++
Social Science	Web Development
English Grammar	Data Science
Commerce	CS Subjects
World GK	