

Recipes

localhost:5000

Register

User Name:

Email:

Dojo Location:  
Chicago

Password:

Confirm Password:

Birth date (must be at least 18 years old):

REGISTER

Sign In

User Name:

Password:

SIGN IN

Validations apply, display error messages.  
All fields must be filled out  
Email must be unique and valid  
Passwords must match  
Passwords must be at least 8 characters  
Birthday must be at least 18 years ago

Recipes

localhost:5000/dashboard

Welcome Kermit!

Create

Click here to create new recipes

Review

Click here to review your existing recipes

User must be logged in to reach this page.

Logout

Recipes

localhost:5000/create

Welcome Kermit!

Dashboard

Review

Logout

Create a recipe

Recipe name:

Description:

Instructions:

Under 30 minutes to prepare?  
☐ Yes ☐ No

CREATE

User must be logged in to reach this page.  
Validations apply, display error messages.  
All fields must be filled out  
Recipe name, description, and instructions must all be at least 3 characters long.

Recipes

localhost:5000/review

Welcome Kermit!

Dashboard

Create

Logout

These are your recipes

Name	Description	Under 30 minutes?	Actions
Spaghetti	Good old spaghetti	Yes	<a href="#">view instructions</a> <a href="#">delete</a>
Cheesecake	Heart attack on a plate	No	<a href="#">view instructions</a> <a href="#">delete</a>

User must be logged in to reach this page.

Clicking the delete link will remove the recipe from the database and be removed from the table.

Recipes

localhost:5000/instructions/27

Welcome Kermit!

Dashboard

Create

Logout

Instructions for spaghetti

Description: Good old spaghetti

Under 30 minutes: Yes

Instructions: Stir diced tomatoes, tomato sauce, and tomato paste into a pan. Season with oregano, basil, salt, and pepper. Turn on the stove. When the sauce burns, throw it out and heat up a can of spaghettiOs.

GO BACK

DELETE

User must be logged in to reach this page.  
Clicking the Go Back button redirects to the user to /review.  
Clicking the Delete button will remove the recipe from the database and redirect the user to /review.

### Server-side code example:

```
from flask import Flask, render_template, redirect, request, session, flash
from mysqlconnection import connectToMySQL as connect

app= Flask(__name__)

@app.route('/dashboard')
def index():
    mysql = connect('books')
    query = "SELECT * FROM books;"
    allbooks = mysql.query_db(query)

    return render_template('dashboard.html', data=allbooks)

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

@app.route('/insertingintodb', methods=['POST'])
def insertingintodb():
    mysql=connect('books')
    query = "INSERT INTO books (tittle, description, created_at, updated_at) VALUES (%(title)s, %(description)s, NOW(),NOW());"
    data = {
        'title' : request.form['title'],
        'description' : request.form['description']
    }
    insert = mysql.query_db(query,data)
    print(insert)
    return redirect('/dashboard')

@app.route('/edit_book/<id>', methods=["POST"])
def edit_book(id):

    mysql = connect('books')
    query = "SELECT * FROM books WHERE id = %(id)s;"
    data = { 'id' : id}
    edit = mysql.query_db(query, data)
    print(edit)

    return render_template('edit.html', books=edit)

@app.route('/edit_process', methods=['POST'])
def edit_process():

    mysql = connect('books')
    query = "UPDATE books SET tittle = %(tittle)s, description = %(description)s WHERE id = %(id)s;"
```

```

data={ 'id' : request.form['bookID'],
      'tittle' : request.form['tittle'],
      'description' : request.form['description']
    }
delete = mysql.query_db(query,data)

return redirect('/dashboard')

@app.route('/delete/<id>', methods=['POST'])
def delete(id):
    mysql = connect('books')
    query = "DELETE FROM books WHERE id = %(id)s"
    data={
        'id':id
    }
    delete = mysql.query_db(query,data)
    return redirect('/dashboard')

```

### Add Book page

## Add Book:

Book Title:

Description:



```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
    crossorigin="anonymous">
  <title>Add Book</title>
</head>
<body>
  <div class="container">
    <h1>Add Book:</h1>
    <form action="/insertingintodb" method="POST">

```

```
<div class="form-group">
  <label >Book Title:</label>
  <input type="text" class="form-control" placeholder="Book Title" name="title">
</div>
<div class="form-group">
  <label >Description:</label>
  <input type="text" class="form-control" placeholder="Description" name="description">
</div>

  <button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>

</body>
</html>
```

### Edit Book page

## Edit Book

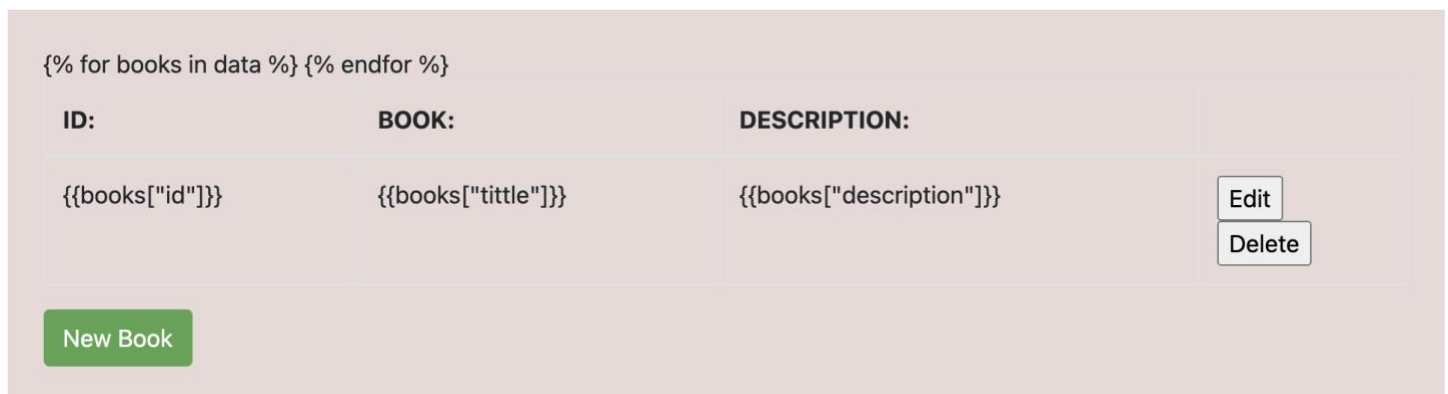
Book Title

Description:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Edit Book</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
integrity="sha384-WskhaSGFgHYWDcbwN70/dfYBj47jz9qbsMIId/iRN3ewGhXQFZCSftd1LZCfmhktB"
  crossorigin="anonymous">
</head>
<body>
  <div class="container">
```

```
<h1>Edit Book</h1>
<form action="/edit_process" method="POST">
  <input type="hidden" name="bookID" value="{{books[0]['id']}}">
  <div class="form-group">
    <label >Book Title</label>
    <input type="text" class="form-control" value="{{books[0]['tittle']}}" name="tittle">
  </div>
  <div class="form-group">
    <label>Description:</label>
    <input type="text" class="form-control" value="{{books[0]['description']}}" name="description">
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>
</body>
</html>
```

### Dashboard page



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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
  integrity="sha384-WskhaSGFgHYWDcbwN70/dfYBj47jz9qbsMIId/iRN3ewGhXQFZCSftd1LZCfmhktB"
  crossorigin="anonymous">
  <title>HTML Table</title>
  <style>
    .container {
      margin: 2%;
```

```

padding: 2%;
background-color: rgb(235, 217, 217);
}

</style>
</head>

<body>
  <div class="container">
    <table class="table table-bordered">
      <thead>
        <tr>
          <th scope="row">ID:</th>
          <th scope="row">BOOK:</th>
          <th scope="row">DESCRIPTION:</th>
        </tr>
      </thead>
      <tbody>
        {% for books in data %}
          <tr>
            <td>
              {{books["id"]}}
            </td>
            <td>
              {{books["tittle"]}}
            </td>
            <td>
              {{books["description"]}}
            </td>
            <td>
              <form action="/edit_book/{{books['id']}}" method="POST">
                <input type="submit" value="Edit">
              </form>
              <form action="/delete/{{books['id']}}" method="POST">
                <input type="submit" value="Delete">
              </form>
            </td>
          </tr>
        {% endfor %}

      </tbody>
    </table>

    <a href="/add_book" class="btn btn-success">New Book</a>

  </div>
</body>

</html>

```

## Login Page

```
{% with messages = get_flashed_messages(category_filter=["permission"]) %} {% if messages %} {% for message in messages %} {{ message }}
{% endfor %} {% endif %} {% endwith %}
```

### Login and Registration

```
{% with messages = get_flashed_messages(category_filter=["register"]) %} {% if messages %} {% for message in messages %} {{ message }}
{% endfor %} {% endif %} {% endwith %}
```

#### Register here!

First name:

This field is required

Last name:

This field is required

Email:

This field is required

Password

This field is required

Confirm Password

This field is required

Register!

#### Login here!

```
{% with messages =
```

```
get_flashed_messages(category_filter=["login"])
%} {% if messages %} {% for message in
messages %} {{ message }}
{% endfor %} {% endif %} {% endwith %}
```

Email:

This field is required

Password

This field is required

Login!

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8">
```

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

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

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css"
integrity="sha384-WskhaSGFgHYWDcbwN70/dfYBj47jz9qbsMIId/iRN3ewGhXQFZCStfd1LZCfmhktB"
crossorigin="anonymous">
```

```
<title>Login and Registration</title>
```

```
<style>
```

```
div.container {
  width: 1080px;
  min-height: 720px;
  background-color: lightgrey;
  margin: 0px auto;
  padding: 5px;
```

```

}
div.registration {
    display: inline-block;
    vertical-align: top;
    width: 335px;
    margin-left: 20px;
    padding: 10px;
}
div.login_here {
    display: inline-block;
    vertical-align: top;
    width: 390px;
    padding: 10px;
    margin-left: 235px;
}

.error {
    color: red;
}
#text {
    margin-left: 50px;
}
</style>

</head>

<body>
<div class="container">
    {% with messages = get_flashed_messages(category_filter=["permission"]) %}
    {% if messages %}
    {% for message in messages %}
    <span class="error">{{ message }}</span><br />
    {% endfor %}
    {% endif %}
    {% endwith %}

    <h2 id="text">Login and Registration</h2>
    {% with messages = get_flashed_messages(category_filter=["register"]) %}
    {% if messages %}
    {% for message in messages %}
    <span class="error">{{ message }}</span><br />
    {% endfor %}
    {% endif %}
    {% endwith %}

    <br>
    <br>

```



```

<div class="registration">
  <h4>Register here!</h4>
  <form action="/register" method="POST">
    <div class="form-group">
      <label for="first_name">First name:</label>
      <!-- <span class="error">This field is required</span> -->
      <input class="form-control is-invalid" type="text" id="first_name" name="first_name"
placeholder="This field is required">
    </div>
    <div class="form-group">
      <label for="last_name">Last name:</label>
      <!-- <span class="error">This field is required</span> -->
      <input class="form-control is-invalid" type="text" id="last_name" name="last_name"
placeholder="This field is required">
    </div>
    <div class="form-group">
      <label for="email">Email:</label>
      <!-- <span class="error">This field is required</span> -->
      <input class="form-control is-invalid" type="text" id="email" name="email" placeholder="This field
is required">
    </div>
    <div class="form-group">
      <label for="password">Password</label>
      <!-- <span class="error">This field is required</span> -->
      <input class="form-control is-invalid" type="password" id="password_hash"
name="password_hash"
placeholder="This field is required">
    </div>
    <!-- <div class="invalid-feedback">

</div> -->
    <div class="form-group">
      <label for="password2">Confirm Password</label>
      <!-- <span class="error">This field is required</span> -->
      <input class="form-control is-invalid" type="password" id="password2" name="password2"
placeholder="This field is required">
      <!-- <div class="invalid-feedback">

</div> -->
    </div>
    <button class="btn btn-success" type="submit">Register!</button>
  </div>
</form>

<div class="login_here">
  <h4>Login here!</h4>
  {% with messages = get_flashed_messages(category_filter=["login"]) %}
  {% if messages %}
  {% for message in messages %}

```

```

    <span class="error">{{ message }}</span><br />
    {% endfor%}
    {% endif %}
    {% endwith %}
    <form action="/login" method="POST">
        <div class="form-group">
            <label for="email">Email:</label>
            <input class="form-control is-invalid" type="text" id="email" name="email" placeholder="This field
is required">
        </div>
        <div class="form-group">
            <label for="password">Password</label>
            <input class="form-control is-invalid" type="text" id="password" name="password"
placeholder="This field is required">
        </div>
        <button class="btn btn-success" type="submit">Login!</button>

    </form>
</div>

</div>

</body>

</html>

```

### Server-side code:

```

from flask import Flask, render_template, request, redirect, session, flash
from mysqlconnection import connectToMySQL
from flask_bcrypt import Bcrypt
import re

app = Flask(__name__)
bcrypt = Bcrypt(app)

EMAIL_REGEX = re.compile(r'^[a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z]+$')
Name_REGEX = re.compile(r'^[a-zA-Z]+$')
app.secret_key = "secret_key"

@app.route('/')
def index():
    if 'user_id' not in session:
        session['user_id'] = None

```

```

return render_template('index.html')

@app.route("/register", methods=['POST'])
def register():

    pw_hash = bcrypt.generate_password_hash(request.form['password_hash'])
    print(pw_hash)

    session.clear()
    errors = False
    for key in request.form:
        flash(request.form[key], 'register')
        if len(request.form[key]) < 1:
            if errors == False:
                flash("Please complete the form!", "register")
                errors = True
    if errors:
        return redirect('/')

    if len(request.form['password_hash']) < 1:
        flash("please complete all required fields prior to submission"
            ) # just pass a string to flash function broski
        return redirect('/')
    if len(request.form['password2']) < 1:
        flash("please complete all required fields prior to submission"
            ) # just pass a string to flash function broski
        return redirect('/')
    if request.form['password2'] != request.form['password_hash']:
        flash("password fields do not match"
            ) # just pass a string to flash function
        return redirect('/')

    mysql = connectToMySQL('registrations')
    query = "SELECT * FROM users WHERE email = %(email)s;"
    data = {"email": request.form['email']}
    results = mysql.query_db(query, data)
    if results:
        flash("This email address is already taken!", "register")
    else:
        mysql = connectToMySQL('registrations')
        query2 = "INSERT INTO users (first_name, last_name, email, password_hash) VALUES(%(first_name)s,
%(last_name)s, %(email)s, %(pw_hash)s);"
        data = {
            'first_name': request.form['first_name'],
            'last_name': request.form['last_name'],
            'email': request.form['email'],
            'pw_hash': pw_hash
        }

```

```

mysql.query_db(query2, data)
mysql = connectToMySQL('registrations')
query = "SELECT user_id FROM users WHERE email = %(email)s;"
session['user_id'] = mysql.query_db(query, data)
return redirect('/success')

# mysql = connectToMySQL('registrations')
# newuserid = mysql.query_db(query2, data)
# if newuserid:
#     session['first_name'] = request.form['first_name']
#     session['user_id'] = newuserid
#     flash("You have been successfully registered!", "register")
#     return redirect('/sucess')
# else:
#     flash("We're sorry, we could not register you at this time",
#         # "register")
return redirect('/success')

@app.route('/login', methods=['POST'])
def login():
    session.clear()
    query = "SELECT * FROM users WHERE email = %(email)s;"
    data = {"email": request.form['email']}
    mysql = connectToMySQL('registrations')
    results = mysql.query_db(query, data)
    if results:
        session['user_id'] = results[0]['user_id']
        session['first_name'] = results[0]['first_name']
        return redirect('/success')
    flash("We're sorry, we could not log you in", "login")
    return redirect("/")

@app.route('/success')
def success_page():
    if 'user_id' not in session or session['user_id'] == None:
        flash("You must be logged in to enter the website", "permission")
        return redirect("/")

    return render_template("success.html")

@app.route('/logout')
def logout():
    session.clear()
    return redirect('/')

```