

Introduction:

Full Stack development in employee data store in mysql.(CRUD) Applications

TEMPLATE:**INDEX .HTML:**

```
{% extends "layout.html" % }
{% block body % }
<div class="row"><h3>Employee</h3></div>
<div class="row">
<div class="col-md-4">
{% with messages = get_flashed_messages() % }
{% if messages % }
{% for message in messages % }
<div class="alert alert-success alert-dismissible fade show" role="alert">
{ { message } }
<button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">×</span>
</button>
</div>
{% endfor % }
{% endif % }
{% endwith % }
<div class="card card-body">
<form action="{ { url_for('add_employee') } }" method="POST">
<div class="form-group">
<input type="text" class="form-control" name="fullname"
placeholder="FullName">
</div>
```

```
<div class="form-group">
```

```
<input type="text" class="form-control" name="email" placeholder="Email">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" class="form-control" name="phone" placeholder="Phone">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" class="form-control" name="work" placeholder="work">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" class="form-control" name="salary" placeholder="salary">
```

```
</div>
```

```
<button class="btn btn-primary btn-block">
```

Save

```
</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div class="col-md-8">
```

```
<table id="example" class="table table-striped table-bordered" style="width:100%">
```

```
<thead>
```

```
<tr>
```

```
<td>ID</td>
```

```
<td>fullname</td>
```

```
<td>email</td>
```

```
<td>phone</td>
```

```
<td>work</td>
<td>salary</td>
<td>Action</td>
</tr>
</thead>
<tbody>
{% for data in employee %}
<tr>
<td>{{ data.id }}</td>
<td>{{ data.name }}</td>
<td>{{ data.email }}</td>
<td>{{ data.phone }}</td>
<td>{{ data.work }}</td>
<td>{{ data.salary }}</td>
<td width="130">
<a href="/edit/{{ data.id }}" class="btn btn-secondary btn-sm">edit</a>
<a href="/delete/{{ data.id }}" class="btn btn-danger btn-delete btn-sm">delete</a>
</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
{% endblock %}
```

LAYOUT.HTML:

```
<!DOCTYPE html>
```

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

```
  <head>
```

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

```
    <title>Employee Details</title>
```

```
  <link
```

```
    href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
```

```
    rel="stylesheet" id="bootstrap-css">
```

```
  <script
```

```
    src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```

```
  <script
```

```
    src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

```
  <script
```

```
    src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
```

```
  <script
```

```
    src="https://cdn.datatables.net/1.10.16/js/dataTables.bootstrap4.min.js"></script>
```

```
> </head>
```

```
  <body>
```

```
    <!-- NAVIGATION -->
```

```
    <nav class="navbar navbar-dark bg-dark">
```

```
      <a class="navbar-brand" href="/">Employee Details</a>
```

```
    </nav>
```

```
    <div class="container pt-4">
```

```
      { % block body % }
```

```
    { % endblock % }
```

</div>

```
<script>          const          btnDelete=
document.querySelectorAll('.btn-delete');
if(btnDelete) {          const  btnArray  =
Array.from(btnDelete);
btnArray.forEach((btn)          =>          {
btn.addEventListener('click',    (e)    =>    {
if(!confirm('Are you sure you want to delete
it?')){
    e.preventDefault();
  }
});
})
}
```

```
$(document).ready(function() {
  $('#example').DataTable({
    "aLengthMenu": [[3, 5, 10, 25, -1], [3, 5, 10, 25, "All"]],
    "iDisplayLength": 3
  }
);
} );
```

</script>

</body>

</html>

EDIT.HTML:

```
{% extends "layout.html" % }
```

```
{% block body % }
```

```
<div class="row">
```

```
<div class="col-md-4 offset-md-4">
```

```
<div class="card card-body">
```

```
<form action="/update/{{ employee.id }}" method="POST">
```

```
<div class="form-group">
```

```
<input type="text" name="fullname" value="{{ employee.name }}"
class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" name="phone" value="{{ employee.phone }}"
class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" name="email" value="{{ employee.email }}"
class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" name="work" value="{{ employee.work }}"
class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" name="salary" value="{{ employee.salary }}"
class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<button type="submit" class="btn btn-primary btn-block">
```

```
    Update
```

```
</button>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
{ % endblock % }
```

App.py:

```
from flask import Flask, render_template, request, redirect, url_for, flash
```

```
from flaskext.mysql import MySQL
```

```
import pymysql
```

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

```
app.secret_key = "Cairocoders-Ednalan"
```

```
mysql = MySQL()
```

```
# MySQL configurations
```

```
app.config['MYSQL_DATABASE_USER'] = 'root'
```

```
app.config['MYSQL_DATABASE_PASSWORD'] = "
```

```
app.config['MYSQL_DATABASE_DB'] = 'testingdb'
```

```
app.config['MYSQL_DATABASE_HOST'] = 'localhost'
```

```
mysql.init_app(app)
```

```
@app.route('/')
```

```
def Index():
```

```
    conn = mysql.connect()
```

```
    cur = conn.cursor(pymysql.cursors.DictCursor)
```

```
    cur.execute('SELECT * FROM employee')
```

```
    data = cur.fetchall()
```

```
    cur.close()
```

```
    return render_template('index.html', employee = data)
```

```
@app.route('/add_contact', methods=['POST'])
```

```
def add_employee():
```

```
    conn = mysql.connect()
```

```
    cur = conn.cursor(pymysql.cursors.DictCursor)
```

```
    if request.method == 'POST':
```

```
        fullname = request.form['fullname']
```

```
        phone = request.form['phone']
```

```
        email = request.form['email']
```

```
        cur.execute("INSERT INTO employee (name, email, phone,work,salary)  
VALUES (%s,%s,%s,%s,%s)", (fullname, email, phone,work,salary))
```

```
        conn.commit()
```

```
        flash('Employee Added successfully')
```

```
        return redirect(url_for('Index'))
```



```
@app.route('/edit/<id>', methods = ['POST', 'GET'])
def get_employee(id):
    conn = mysql.connect()
    cur = conn.cursor(pymysql.cursors.DictCursor)

    cur.execute('SELECT * FROM employee WHERE id = %s', (id))
    data = cur.fetchall()
    cur.close()
    print(data[0])
    return render_template('edit.html', employee = data[0])
```

```
@app.route('/update/<id>', methods=['POST'])
def update_employee(id):
    if request.method == 'POST':
        fullname = request.form['fullname']
        phone = request.form['phone']
        email = request.form['email']
        work = request.form['work']
        salary = request.form['salary']
        conn = mysql.connect()
        cur = conn.cursor(pymysql.cursors.DictCursor)
        cur.execute("""
            UPDATE employee
            SET name = %s,
                email = %s,
                phone = %s,
```

```
        work = %s,  
        salary = %s  
    WHERE id = %s  
    """, (fullname, email, phone, id))  
    flash('Employee Updated Successfully')  
    conn.commit()  
    return redirect(url_for('Index'))
```

```
@app.route('/delete/<string:id>', methods = ['POST','GET'])  
def delete_employee(id):  
    conn = mysql.connect()  
    cur = conn.cursor(pymysql.cursors.DictCursor)  
  
    cur.execute('DELETE FROM employee WHERE id = {0}'.format(id))  
    conn.commit()  
    flash('Employee Removed Successfully')  
    return redirect(url_for('Index'))
```

```
# starting the app  
if __name__ == "__main__":  
    app.run(port=3306, debug=True)
```

COMMAND PROMPT:

```
#pip install -U
```

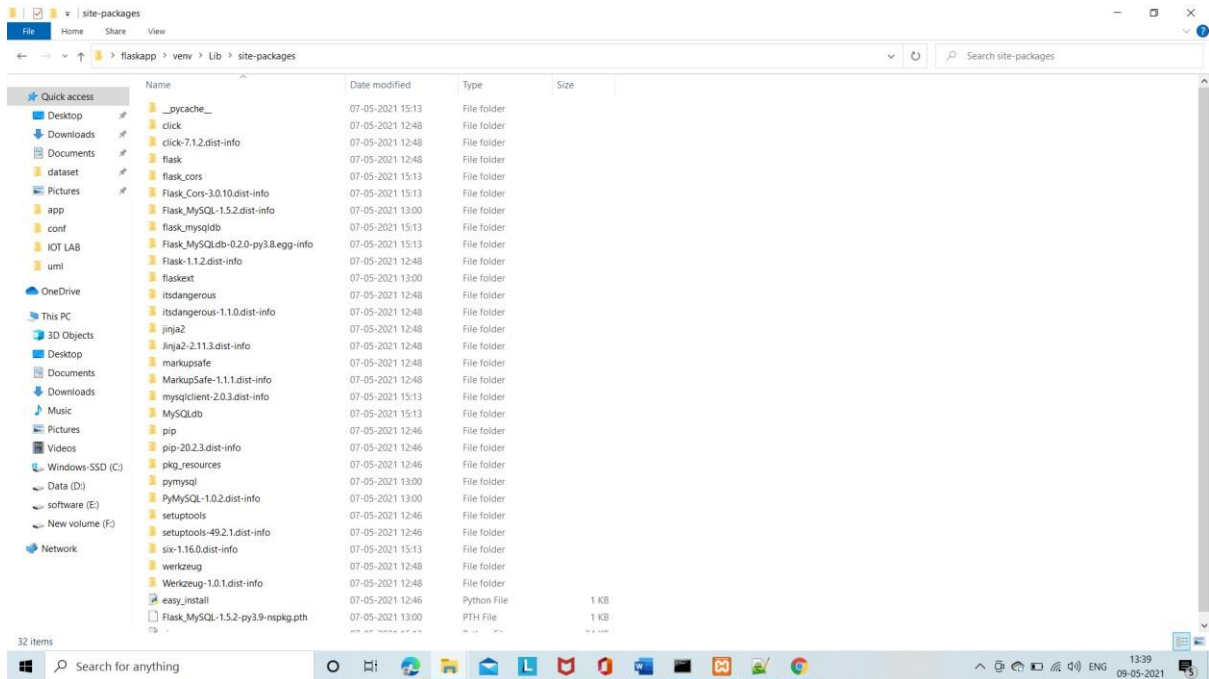
```
flaskcors
```

Venv install:

```
Py -3 -m venv venv
```

```
C:\Users\91876>cd C:\Users\91876\Desktop\flaskapp
C:\Users\91876\Desktop\flaskapp>venv\Scripts\activate
(venv) C:\Users\91876\Desktop\flaskapp>cd app
(venv) C:\Users\91876\Desktop\flaskapp\app>flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [09/May/2021 13:25:03] "[37mGET / HTTP/1.1-[0m" 200 -
127.0.0.1 - - [09/May/2021 13:25:04] "[33mGET /favicon.ico HTTP/1.1-[0m" 404 -
127.0.0.1 - - [09/May/2021 13:26:59] "[37mGET / HTTP/1.1-[0m" 200 -
127.0.0.1 - - [09/May/2021 13:27:02] "[37mGET / HTTP/1.1-[0m" 200 -
127.0.0.1 - - [09/May/2021 13:27:03] "[37mGET / HTTP/1.1-[0m" 200 -
(venv) C:\Users\91876\Desktop\flaskapp\app>http://127.0.0.1:5000/
```

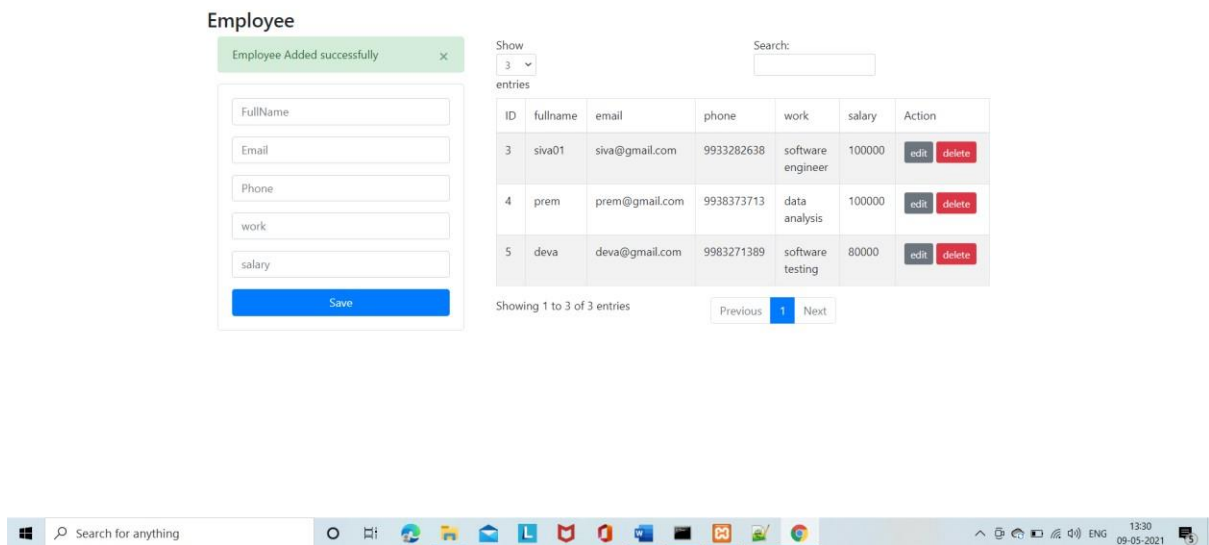
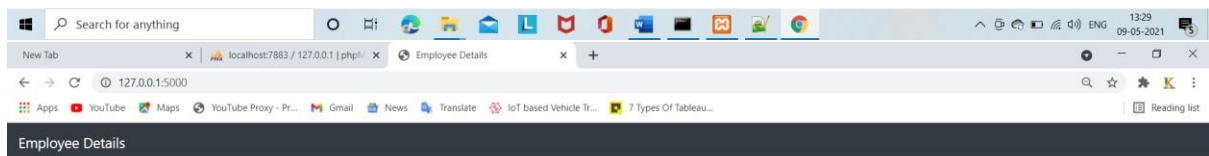
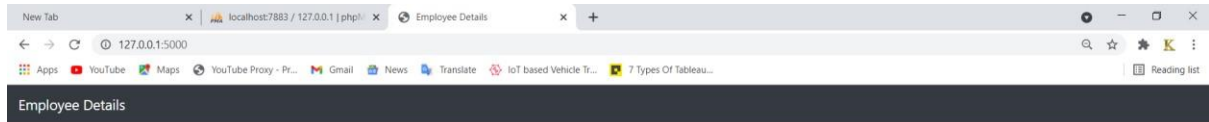
Venv library:



Rest client:

Output:

Insert:



Update:

Employee Details

deva R

9983271389

deva@gmail.com

software testing

90000

Update

Employee Details

Employee

Employee Updated Successfully

FullName

Email

Phone

work

salary

Save

Show 3 entries

ID	fullname	email	phone	work	salary	Action
3	siva01	siva@gmail.com	9933282638	software engineer	100000	edit delete
4	prem	prem@gmail.com	9938373713	data analysis	100000	edit delete
5	deva R	deva@gmail.com	9983271389	software testing	90000	edit delete

Showing 1 to 3 of 3 entries

Previous 1 Next

Delete:

Employee Details

Employee Updated Successfully

127.0.0.1:5000 says
Are you sure you want to delete it?

OK Cancel

Employee

Employee Updated Successfully

FullName

Email

Phone

work

salary

Save

Show 3 entries

Search:

ID	fullname	email	phone	work	salary	Action
3	siva01	siva@gmail.com	9933282638	software engineer	100000	edit delete
4	prem	prem@gmail.com	9938373713	data analysis	100000	edit delete
5	deva R	deva@gmail.com	9983271389	software testing	90000	edit delete

Showing 1 to 3 of 3 entries

Previous 1 Next

127.0.0.1:5000/delete/4

Search for anything

Employee Details

Employee

Employee Removed Successfully

FullName

Email

Phone

work

salary

Save

Show 3 entries

Search:

ID	fullname	email	phone	work	salary	Action
3	siva01	siva@gmail.com	9933282638	software engineer	100000	edit delete
5	deva R	deva@gmail.com	9983271389	software testing	90000	edit delete

Showing 1 to 2 of 2 entries

Previous 1 Next

Search for anything

1332 09-05-2021

The screenshot displays the phpMyAdmin web interface in a browser window. The address bar shows the URL: `localhost:7883/phpmyadmin/index.php?route=/sql&server=1&db=testingdb&table=employee&pos=0`. The interface is divided into several sections:

- Left Sidebar:** Contains a tree view of the database structure, including 'information_schema', 'mysql', 'performance_schema', 'phpmyadmin', 'test', 'testingdb', and 'user'.
- Top Navigation Bar:** Includes tabs for 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', 'Tracking', and 'Triggers'.
- Main Content Area:**
 - A status bar at the top indicates 'Showing rows 0 - 1 (2 total, Query took 0.0009 seconds)'.
 - A SQL query editor shows the query: `SELECT * FROM 'employee'`.
 - Below the query editor are controls for 'Show all', 'Number of rows' (set to 25), 'Filter rows', and 'Sort by key'.
 - A table of results is displayed with columns: `id`, `name`, `email`, `phone`, `work`, and `salary`.
 - Below the table are controls for 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'.
 - A 'Query results operations' section includes links for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.
 - A 'Bookmark this SQL query' section includes a 'Label' input field and a checkbox 'Let every user access this bookmark'.
- Bottom Console:** A small area at the bottom of the interface for displaying messages or errors.

id	name	email	phone	work	salary
3	siva01	siva@gmail.com	9933282638	software engineer	100000
5	deva R	deva@gmail.com	9983271389	software testing	90000