



f



?3 03:02:14



.ite

## Frasco SQLite

O frasco nos permite usar o SQLite3, que é um dos módulos python. É usado para criar aplicativos web de banco de dados. Para entender isso, vamos criar um aplicativo CRUD.



00:02 / 00:25





tc e essas tabelas contêm informações sobre os alunos.

dados chamado **estudante.db** e "aluno" de mesa no SQLite python dado abaixo.

S:

```
nect("student.db")
pened successfully")
te table Student (id INTEGER PRIMARY KEY AUTOINCREMENT, name
ted successfully")
```

l aplicativo crud no Frasco, temos que focar nas funções de  
dice() e controladores para salvar esses dados no banco de dados.

```
plate("index.html");
```

N o, o **índice()**, o método nomeado, é uma função de exibição que  
e: (\). Ele exibe o índice de modelo.html.

E: ice é o homepage do aplicativo crud. Ele contém os vários links  
Usernho-los, podemos adicionar, excluir e visualizar os dados salvos no banco de dados.

índice.html

00:02 / 00:25





```
o the website</h2>
d Employee</a><br><br>
List Records</a><br><br>
>Delete Record</a><br><br>
```

O exibição **está** associado à URL ([/add](#)). Ele torna o modelo  
a abaixo. Ele fornece um formulário básico para inserir as  
in .

a

```
t</title>

rmation</h2>
savedetails" method="post">

<td><input type="text" name="name"></td></tr>
><td><input type="email" name="email"></td></tr>

```

Todas as informações inseridas pelo administrador são enviadas para a URL "salvar detalhes" que está associada aos mét para os dados inseridos pelo adminis aluno. Ele também produziu várias m algum erro ocorrido.



```
@app.route("/savedetails",methods
def saveDetails():
msg = "msg"
if request.method == "POST":
    try:
```

00:02 / 00:25





uccessfully Added"

add the student to the list"

plate("success.html", msg = msg)



El .html sucesso no qual uma mensagem é exibida para o  
a também contém um link para visualizar todos os registros de entrada  
p



etails</title>

{ {msg} }</h3>  
w">View Employees</a>

O método **delete()** está associado à **URL/exclusão**. Ele exibe o modelo **de exclusão.html** que fornece um formulário para o administrador, que solicita entrar no **student\_id** para excluir esse registro daquele aluno específico. Ele também contém um link para visualizar todos os registros de entrada pelos usuários.



## Apague.html

```
<!DOCTYPE html>
<html>
<head>
    <title>delete record</title>
</head>
<body>
    <h3>Remove Student from the
    <form action="/deleterecord" method="post">
```

00:02 / 00:25





O pelo administrador é postado no URL **/deleterecord** que contém o  
co stabelecer a conexão com o banco de dados e, em seguida, excluir  
e: o do Estudante especificada. O URL **/deleterecord** está associado  
a: **ord()** que é dado abaixo.

```
|     record", methods = ["POST"])
|     ) :
|     ["id"]
|     ect("student.db") as con:

|     )
|     te from Student where id = ?", id)
|     cessfully deleted"

|     deleted"

|     plate("delete_record.html", msg = msg)
```

O **ord()** gera uma mensagem dependendo da condição se os dados  
sócesso ou qualquer erro ocorrido. Ele exibiu um modelo HTML  
que mostra uma mensagem para o administrador.

d

```
<html>
<head>
    <title>delete record</title>
</head>
<body>
<h3>{ {msg} }</h3>
<a href="/view">View List</a>
</body>
</html>
```

O modelo `delete_record.html` contém

do Student para o administrador.

Está associado à **visualização do método**, que busca todos os

dados, busca todas as informações e p

00:02 / 00:25



```
        select * from Employees")
tchall()
_template("view.html",rows = rows)
( view.html ) that displays all the information on the browser
```



vi

```
title>

rmation</h3>
```

```
s %}

        </td>
    } }</td>
] } }</td>
s" ] } }</td>

</tr>
{ % endfor %
</table>
<br><br>
<a href="/">Go back to home page</a>
</body>
</html>
```

O script python completo é dado abaixo:

### crud.py

```
from flask import *
import sqlite3
app = Flask(__name__)
@app.route("/")
```

00:02 / 00:25





```
== "POST":\n\n    rm["name"]\n    orm["email"]\n    .form["address"]\n    ect("Student.db") as con:\n    )\n    RT into Student (name, email, address) values (?,?,?,?)", (name,\n\n        ccessfully Added"\n\n        add the Student to the list"\n\n        plate("success.html",msg = msg)\n\n        ") \n\n        nect("Student.db")\n        sqlite3.Row\n        )\n        ct * from Student")\n        ll()\n        plate("view.html",rows = rows)\n        te")\n\n        plate("delete.html")\n        -+-----+----terecord",methods = ["POST"])\ndef deleterecord():\n    id = request.form["id"]\n    with sqlite3.connect("Student.db") as con:\n        try:\n            cur = con.cursor()\n            cur.execute("delete from Student")\n            msg = "record successfully deleted"\n        except:\n            msg = "can't be deleted"\n        finally:\n            return render_template("delete_r\n        if __name__ == "__main__":\n            app.run(debug = True)
```

00:02 / 00:25

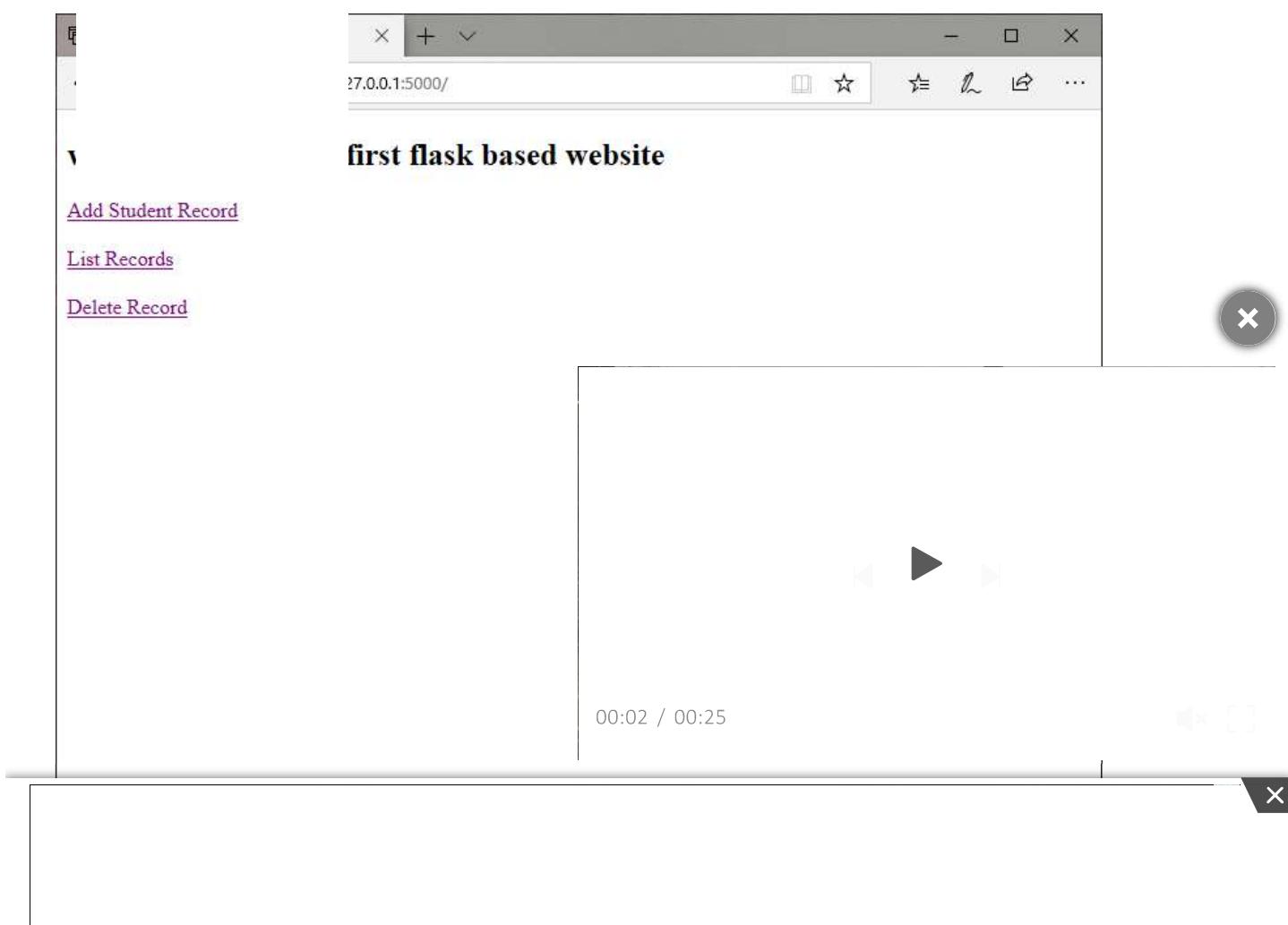
Agora execute o servidor e execute os

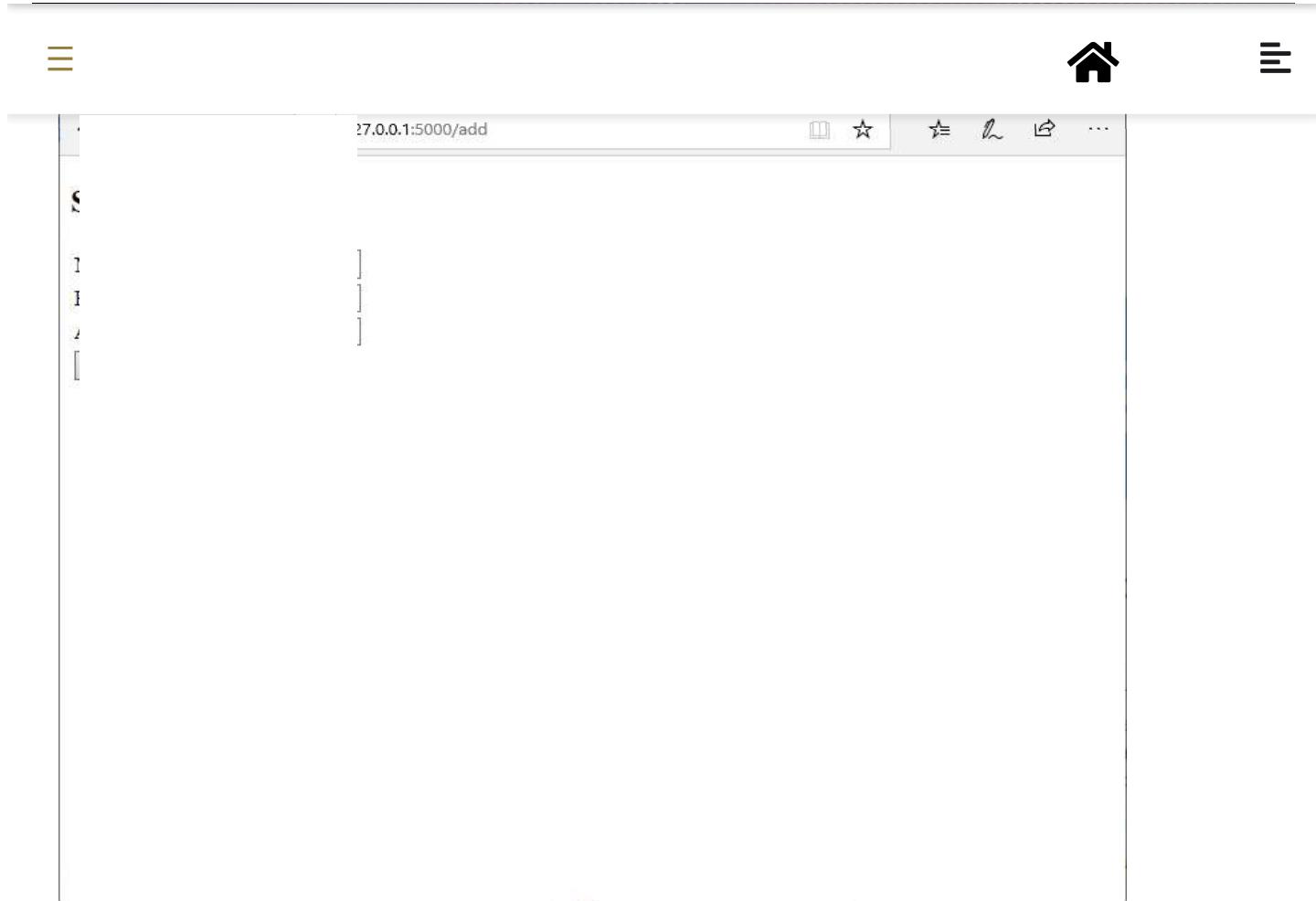


```
C:\> flask_app
:C:\>sk_app>py -m venv env
:C:\>sk_app>app\scripts\activate
:specified.
:C:\>App\scripts\activate
:specified.
:C:\>App\scripts\activate
:specified.
:C:\>env\scripts\activate
:s\flask_app>set flask_app=crud.py
:s\flask_app>flask run
:server. Do not use it in a production deployment.
instead.
:0/ (Press CTRL+C to quit)
```

Agora copie o endereço de URL do cmd e cole-o na seção url no navegador da Web, clique em OK.

Só

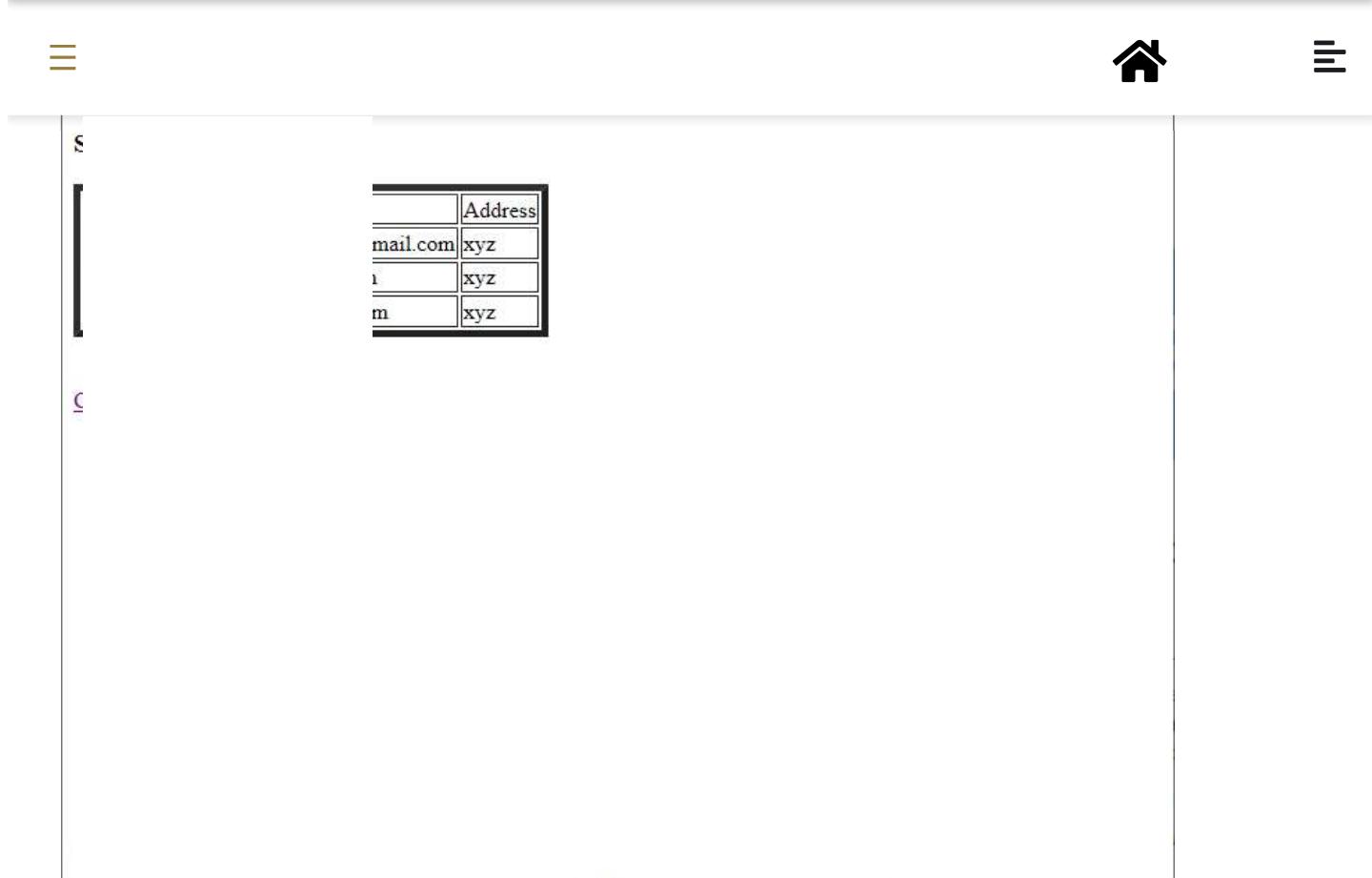




Preencha os campos com os dados do aluno e clique no botão **enviar**.

A screenshot of a web browser window. The address bar shows the URL "27.0.0.1:5000/savedetails". The main content area displays a success message: "Hi Admin, Student successfully saved". Below the message is a link "View Students". A circular close button with an 'X' is visible in the bottom right corner of the content area. The browser has a standard top navigation bar with icons for refresh, search, and other functions. A video player control bar is overlaid at the bottom, showing a play button, a progress bar from 00:02 to 00:25, and other video controls.

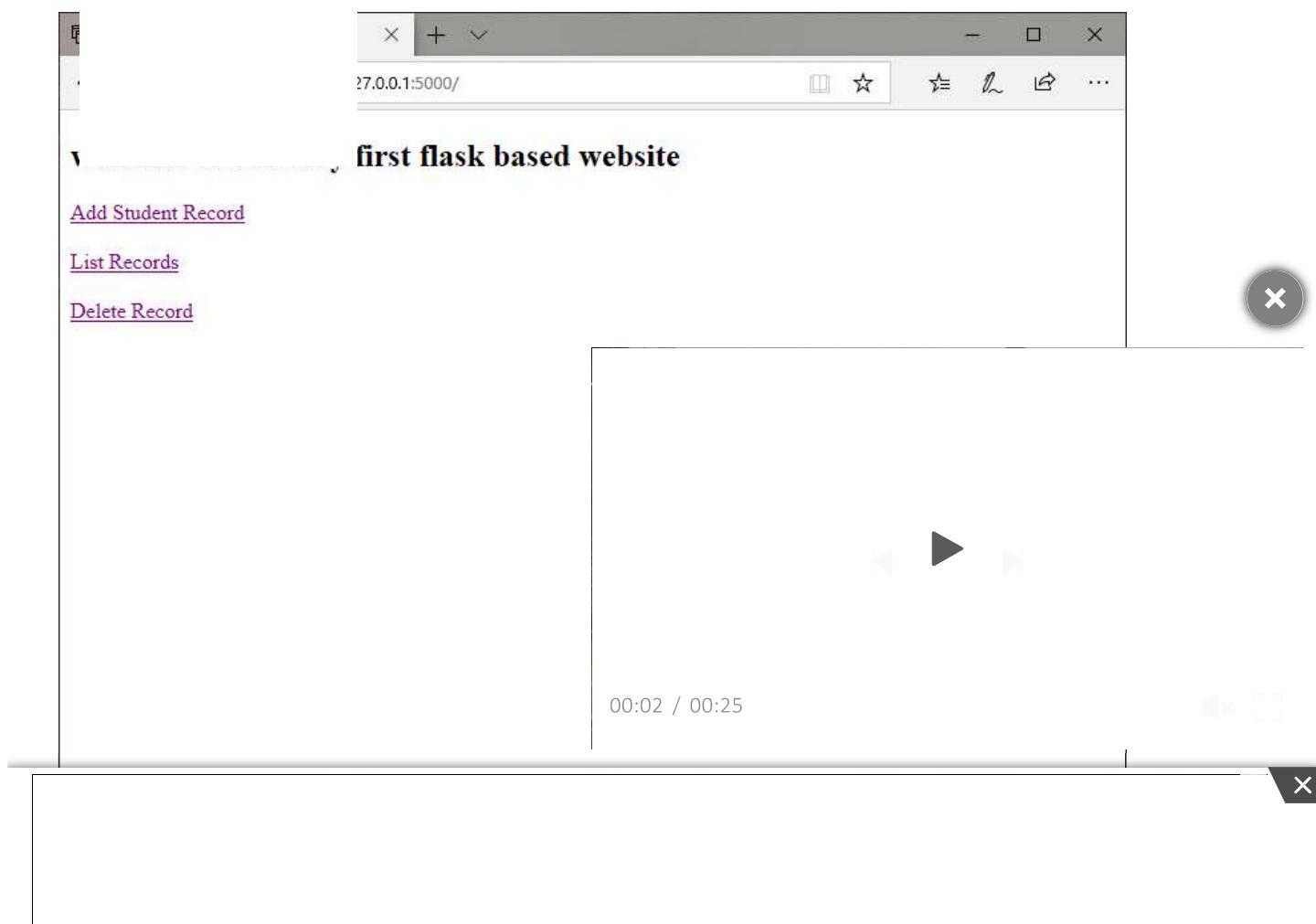
Agora clique no Ver Estudantes para ver os resultados.



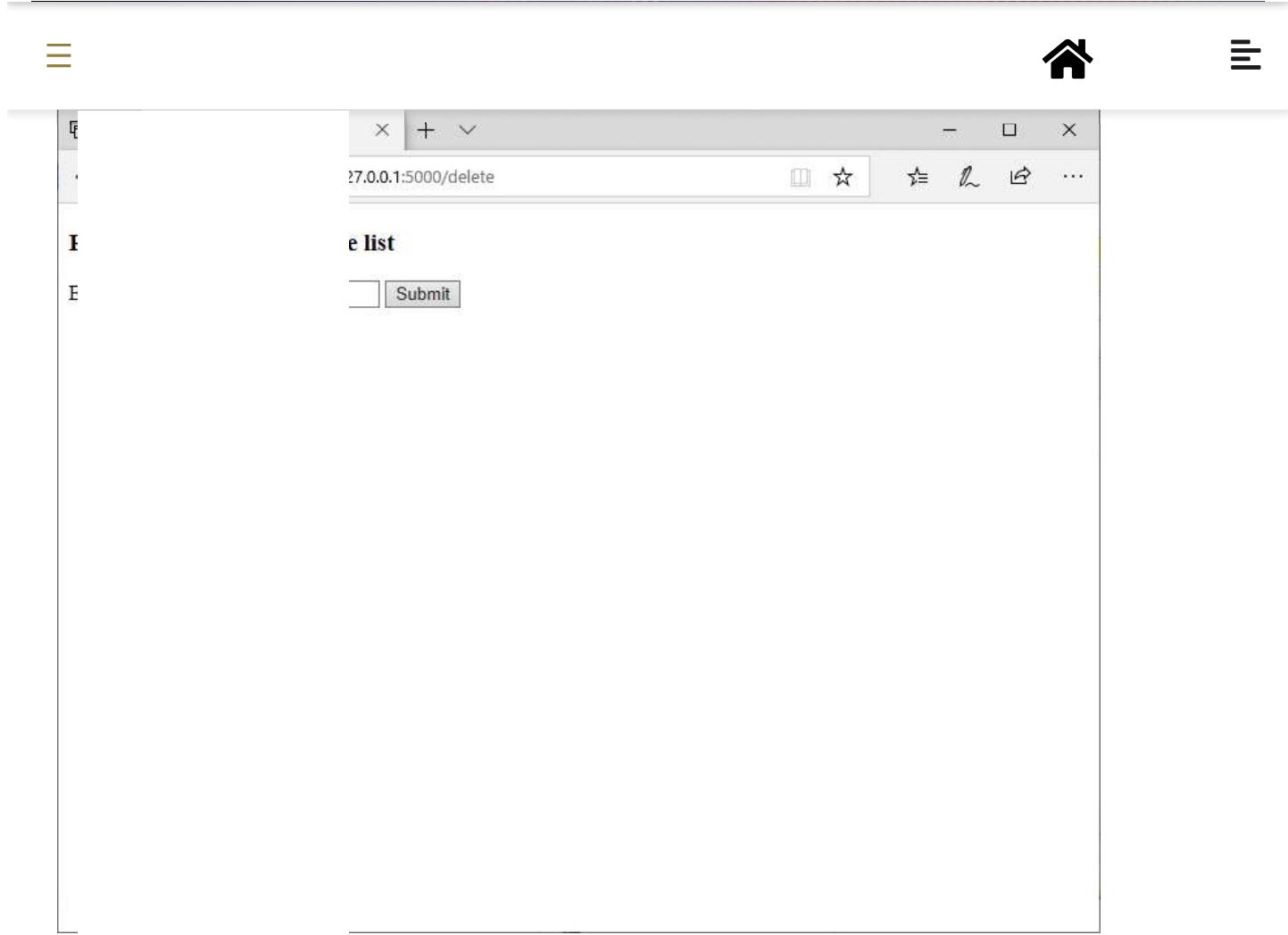
A screenshot of a mobile application interface. At the top, there are three icons: a menu icon (three horizontal bars), a home icon (a house), and a settings icon (three horizontal bars). The main area contains a table with three rows and two columns. The columns are labeled "Address" and "xyz". The rows are labeled "mail.com", "l", and "m".

	Address
mail.com	xyz
l	xyz
m	xyz

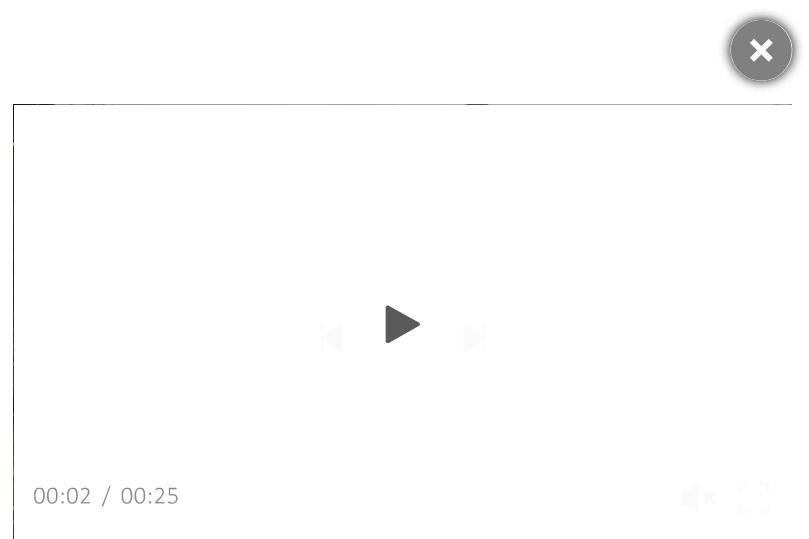
C “**á**gina inicial” para navegar na página inicial.

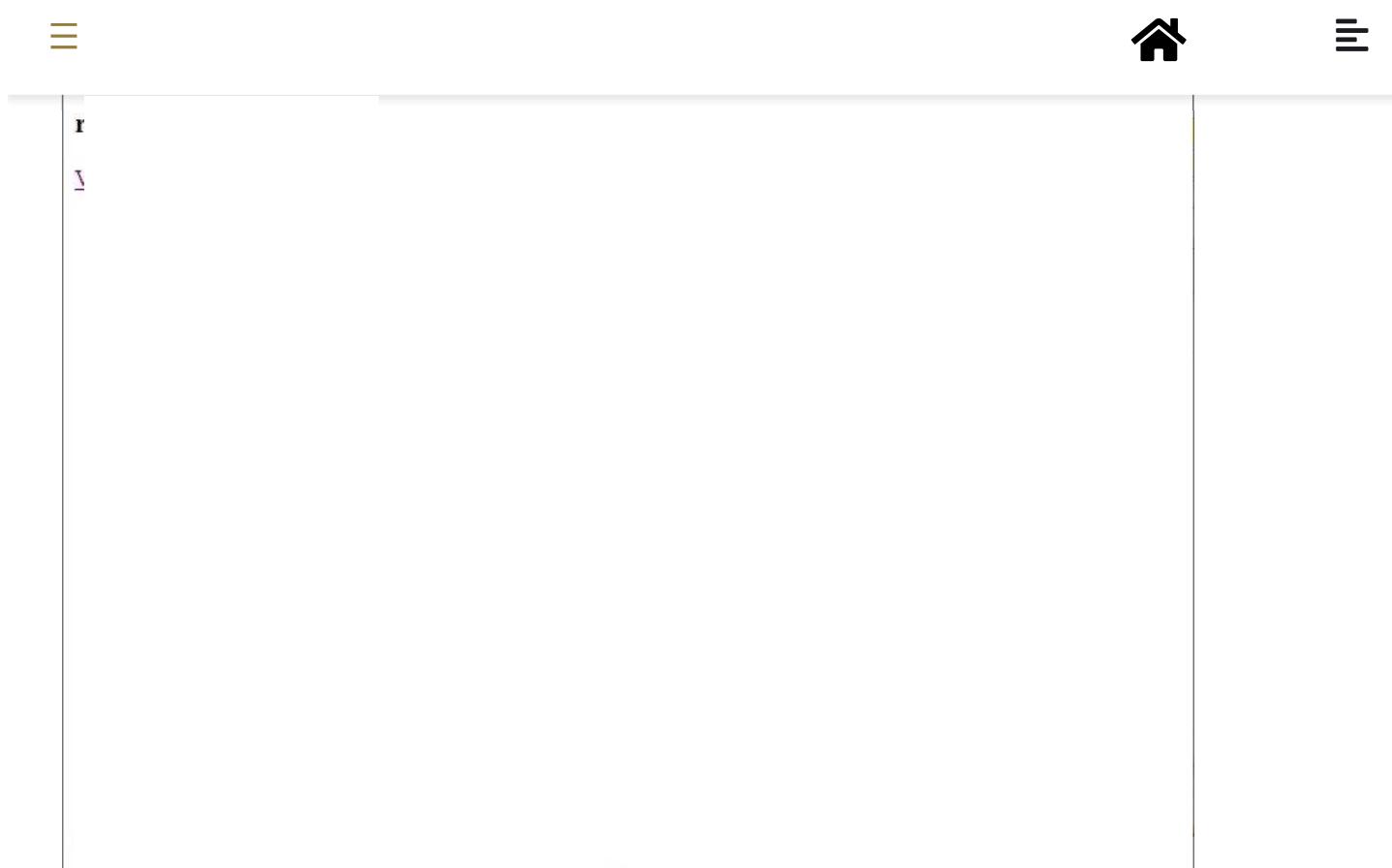


A screenshot of a web browser window. The address bar shows the URL `127.0.0.1:5000/`. The page content is titled "first flask based website". It includes three links: Add Student Record, List Records, and Delete Record. There is also a circular close button with an "X" in the top right corner. A video player control at the bottom indicates the video is at 00:02 / 00:25.



Ação de estudante para excluir o registro desse aluno específico e cl.





P: A operação de exclusão é executada com sucesso, clique no botão  
E:

A screenshot of a web browser window showing a table of student information. The table has columns for ID, Name, Email, and Address. The data is as follows:

ID	Name	Email	Address
2	Akash deep	akashacker0@gmail.com	xyz
4	Roli	xyz1@gmail.com	xyz

[Go back to home page](#)

00:02 / 00:25

The browser interface includes a title bar with a URL (27.0.0.1:5000/view), a toolbar with various icons, and a close button in the top right corner of the main content area.



## Tutc

Tutc

Con le

frasc

Prin

Rote le

frasc

Con o

Mét

Mod

Obj :co

Biscoitos de frasco

Sessão de Frascos

Upload de arquivo de frasco

Redirecionamento de frascos e  
erros

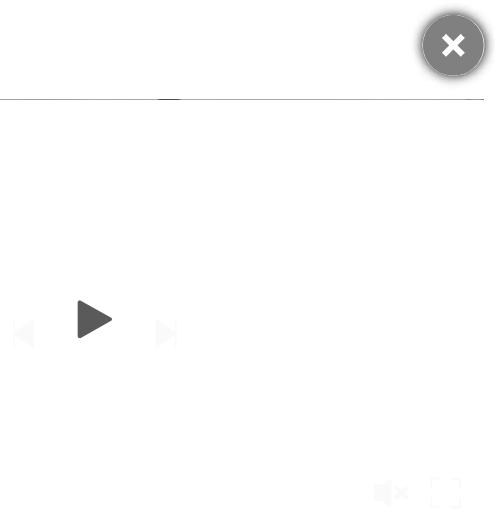
Flash de mensagem de frasco

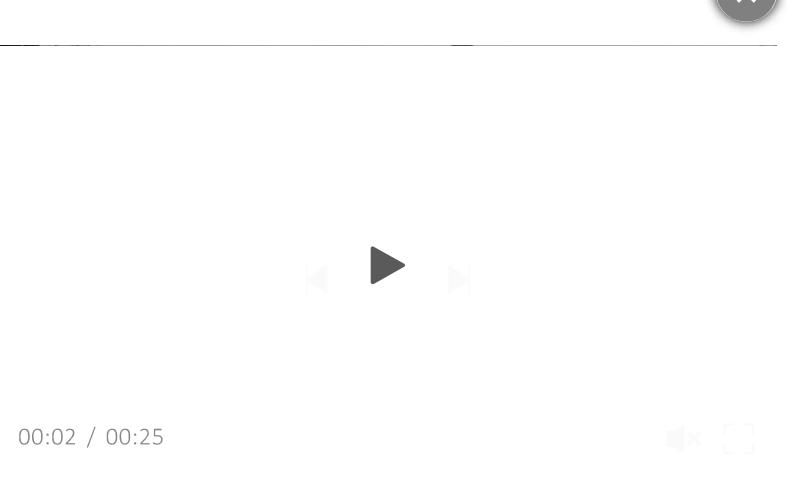
Extensão de correio de frasco

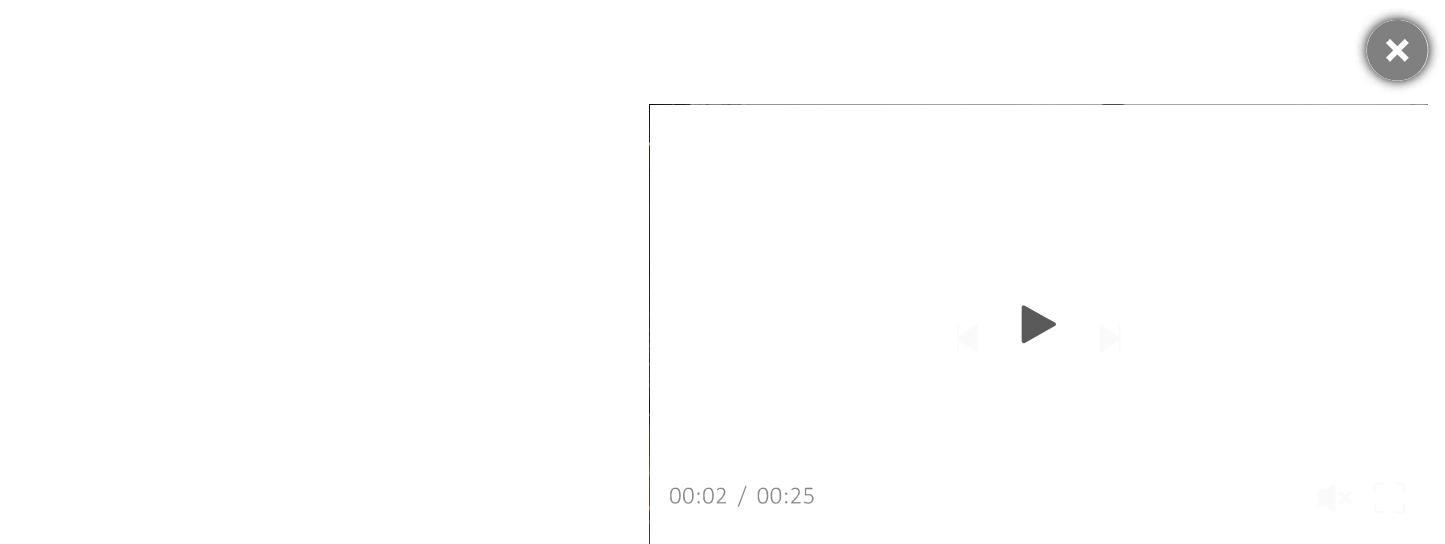
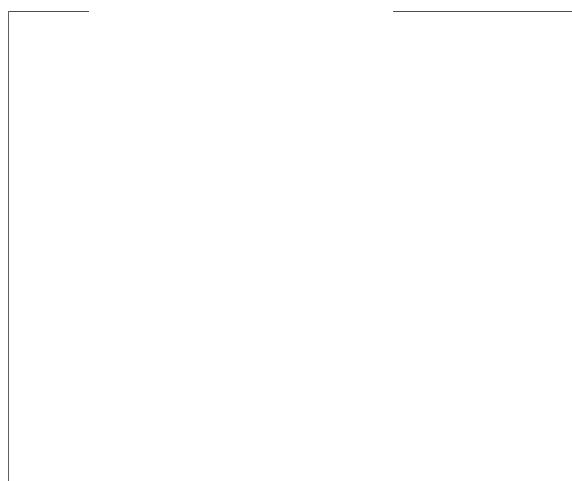
Frasco SQLite

Verificação de e-mail de frasco

00:02 / 00:25









00:02 / 00:25





outadores

ção

es

r

putador

are

:

EVD.js

Springboot

DevOps



Estivador

Kubernet

Aprendizado Profundo

MCQ

DBMS MCQ

McQ de Estrutura de Dados



00:02 / 00:25





esastres

MCQ

MCQ

em

© Copyright 2022 [Tutorialspoint](#)

00:02 / 00:25

