Meet Pithadia

60004210048/B4

Python Programming

Problem Statement: Make a To-Do list app using flask.

Team Members:

Rachita Samant – 60004210107

Tahera Ansari – 60004210115

Abstract:

To- Do list is a list of tasks you need to complete or things that you want to do. Having a list of everything you need to do written down in one place means you shouldn't forget anything important. By prioritising the tasks in the list you plan the order in which you're going to do them and can quickly see what needs your immediate attention and what tasks you can leave until a little later. One of the most important reasons you should use a to do list is that it will help you stay organised. When you write all your tasks in a list, they seem more manageable. When you've got a clear outline of the tasks you've got to do and those you've completed, it helps you stay focused. While freeing up space in your mind for other more creative tasks. When you complete a task, you can cross it off your list. This gives you a sense of progress and achievement, something you'll lack if you're always rushing from one task to the next. If you feel a sense of achievement, it spurs you on and motivates you to keep moving forward.

Here we implemented the to-do list in python using flask framework. We also made use of database for all the crud operations.

Functionalities of App:

In this to-do list app there are 3 major functionalities:

1. Adding the task
2. Updating the task
3. Deleting the task

We can add all the tasks in the list. After adding the tasks, we can either update or delete the tasks. In update we can modify the tasks in the list and delete will help us delete the task once it is done. Apart from these functionalities this to-do list helps us organise our tasks. We can plan our day before through this app. The Simple GUI helps us in easier assess to the to-do list. The minimum functions help us stay focus on our tasks. The app is convenient to use and easier to understand.

Implementation:

Flask:

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

SQL:

Structured Query Language, abbreviated as SQL, is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

Code:

```
from flask import Flask, redirect, render_template, request,url_for
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
# run_with_ngrok(app)


app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///todo.db'

db = SQLAlchemy(app)
task_list = []
class ToDo(db.Model):
        id = db.Column(db.Integer,primary_key=True)
        content = db.Column(db.String(120),  nullable=False)

        def __repr__(self):
            return '%s' %self.content

@app.route('/',methods=['GET','POST'])
def add():
    if request.method == 'POST' :
        task_content = request.form['content']
        task = ToDo(content=task_content)
        db.session.add(task)
        db.session.commit()
        return redirect('/')
    else:
        task_list = ToDo.query.all()
        return render_template('index.html',task_list=task_list)

@app.route('/delete/<int:id>')#02D49845s wswawa
def delete(id):
    task_to_delete = ToDo.query.get_or_404(id)
    db.session.delete(task_to_delete)
    db.session.commit()
```

```
    return redirect('/')

@app.route('/update/<int:id>',methods=['GET','POST'])
def update(id):
    print("hey")
    task_to_updated= ToDo.query.get_or_404(id)
    if request.method == 'POST':
        task_to_updated.content = request.form['update_content']
        db.session.commit()
        return redirect('/')
    else:
        return render_template('update.html',task=task_to_updated)

if __name__ == "__main__":

    app.run(port=5000)
```

Results:

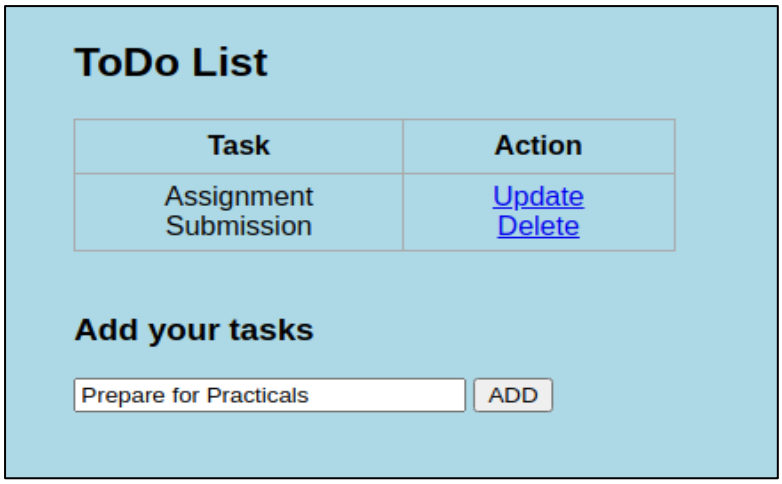

Figure 1: Adding tasks in the list

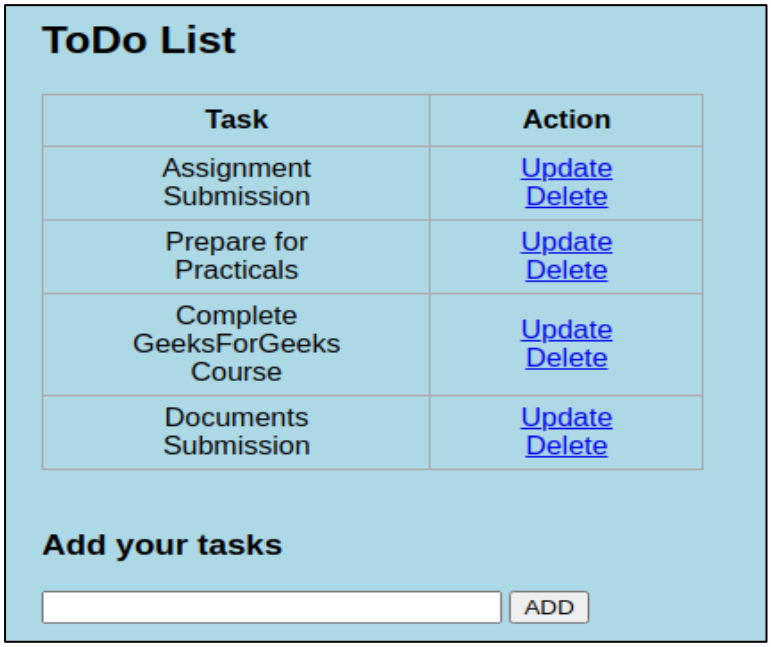

Figure 2: View of task list

Figure 3: Updating the tasks in the list

Figure 4: Deleting the tasks in the list

Conclusion: We created the To-Do list app using flask in python, with different functionalities and connected it with database.