

# Python To-Do Application

## Introduction

The Python To-Do Application is a simple task management tool that allows users to add, view, update, and delete tasks. It is built using Python and can be used as a command-line or web-based application.

## Features

- Add tasks with a title and description
- View the list of pending tasks
- Mark tasks as completed
- Delete tasks
- Save tasks persistently using a database or file storage

## Prerequisites

- Python 3.x
- Required libraries (install using `requirements.txt` if available)

## Installation

1. Create html files

### Index.html

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.rtl.min.css
" integrity="sha384-
```

dpuaG1suU0eT09tx5plTaGMLBsFDLzUCCUXOY2j/LSvXYuG6Bqs43ALlhIqAJVRb"  
crossorigin="anonymous">

```
<title>Demo-site</title>
</head>
<body>
  <nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Navbar</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page"
href="/">MYTODO</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Link</a>
          </li>
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" role="button" data-
bs-toggle="dropdown" aria-expanded="false">
              Dropdown
            </a>
            <ul class="dropdown-menu">
              <li><a class="dropdown-item" href="#">Action</a></li>
              <li><a class="dropdown-item" href="#">Another action</a></li>
              <li><hr class="dropdown-divider"></li>
              <li><a class="dropdown-item" href="#">Something else
here</a></li>
            </ul>
          </li>
          <li class="nav-item">
            <a class="nav-link disabled" aria-disabled="true">Disabled</a>
          </li>
        </ul>
        <form class="d-flex" role="search">
          <input class="form-control me-2" type="search" placeholder="Search"
aria-label="Search">
```

```

        <button class="btn btn-outline-success"
type="submit">Search</button>
    </form>
</div>
</div>
</nav>
<div class="container">
    <form action="/" method="post">
        <div class="mb-3">
            <label for="Todo" class="form-label">Todo title</label>
            <input type="text" class="form-control" id="Todo"
placeholder="Enter your title" name="title">
        </div>
        <div class="mb-3">
            <label for="desc" class="form-label">Description</label>
            <input type="text" id="desc" class="form-control" name="desc"
placeholder="Enter your description">
        </div>
        <div class="text-center">
            <button type="submit" class="btn btn-primary mt-
3">Submit</button>
        </div>
    </form>
</div>
<div class="container">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">sno</th>
                <th scope="col">title</th>
                <th scope="col">desc</th>
                <th scope="col">Time</th>
            </tr>
        </thead>
        <tbody>
            {% for todo in allTodo %}
            <tr>
                <th scope="row">{{loop.index}}</th>
                <td>{{todo.title}}</td>
                <td>{{todo.des}}</td>
                <td>{{todo.date_created}}</td>
                <td>
                    <a href="/delete/{{todo.sno}}" class="btn btn-outline-dark
btn-sm mx-2">Delete</a>

```

```

        <a href="/update/{{todo.sno}}" class="btn btn-outline-dark
btn-sm mx-2">Update</a>
    </td>
</tr>
</tbody>
</table>
</div>

<!-- Bootstrap JS -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
</body>
</html>

```

## Update.html

```

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.rtl.min.css
" integrity="sha384-
dpuaG1suU0eT09tx5p1TaGMLBsfdLzUCCUXOY2j/LSvXYuG6Bqs43ALlhIqAJVRb"
crossorigin="anonymous">
    <title>Update Todo</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Navbar</a>

```

```

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
            <li class="nav-item">
                <a class="nav-link active" aria-current="page"
href="/">MYTODO</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Link</a>
            </li>
            <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle" href="#" role="button" data-
bs-toggle="dropdown" aria-expanded="false">
                    Dropdown
                </a>
                <ul class="dropdown-menu">
                    <li><a class="dropdown-item" href="#">Action</a></li>
                    <li><a class="dropdown-item" href="#">Another action</a></li>
                    <li><hr class="dropdown-divider"></li>
                    <li><a class="dropdown-item" href="#">Something else
here</a></li>
                </ul>
            </li>
            <li class="nav-item">
                <a class="nav-link disabled" aria-disabled="true">Disabled</a>
            </li>
        </ul>
        <form class="d-flex" role="search">
            <input class="form-control me-2" type="search" placeholder="Search"
aria-label="Search">
            <button class="btn btn-outline-success"
type="submit">Search</button>
        </form>
    </div>
</div>
</nav>
<div class="container mt-5">
    <h2>Update Todo</h2>
    <form action="/update/{{ todo.sno }}" method="POST">
        <div class="mb-3">

```

```

        <label for="title" class="form-label">Todo Title</label>
        <input type="text" class="form-control" id="title" name="title"
value="{{ todo.title }}" required>
    </div>
    <div class="mb-3">
        <label for="desc" class="form-label">Description</label>
        <input type="text" id="desc" class="form-control" name="desc"
value="{{ todo.des }}" required>
    </div>
    <button type="submit" class="btn btn-primary">Update</button>
    <a href="/" class="btn btn-secondary">Cancel</a>
</form>
</div>

<!-- Bootstrap JS -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
YvpcrYf0tY3lHB60NNkm6F9B9DVZLESaAA55NDz0xhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
</body>
</html>

```

2. Create a virtual environment (optional but recommended): `python -m venv venv`  
`source venv/bin/activate` # On Windows use ``venv\Scripts\activate``
3. Install dependencies: `pip install -r requirements.txt`
4. Create app.py file

## app.py

```

from flask import Flask, render_template, request, redirect
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
import os

# Initialize Flask app
app = Flask(__name__)

```

```

# Configure SQLite Database in the Current Working Directory
BASE_DIR = os.path.abspath(os.getcwd()) # Get current working directory
app.config['SQLALCHEMY_DATABASE_URI'] = f"sqlite:///{BASE_DIR}/todo.db"
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

# Initialize Database
db = SQLAlchemy(app)

# Define Todo Model
class Todo(db.Model):
    sno = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(200), nullable=False)
    des = db.Column(db.String(500), nullable=False)
    date_time = db.Column(db.DateTime, default=datetime.utcnow)

# Home Route - List all todos & Add a new one
@app.route("/", methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        title = request.form['title']
        desc = request.form['desc']
        if title and desc:
            todo = Todo(title=title, des=desc)
            db.session.add(todo)
            db.session.commit()
    allTodo = Todo.query.order_by(Todo.date_time.desc()).all()
    return render_template('index.html', allTodo=allTodo)

# Delete Todo Route
@app.route('/delete/<int:sno>')
def delete(sno):
    todo = Todo.query.get_or_404(sno)
    db.session.delete(todo)
    db.session.commit()
    return redirect("/")

# Update Todo Route
@app.route('/update/<int:sno>', methods=['GET', 'POST'])
def update(sno):
    todo = Todo.query.get_or_404(sno)
    if request.method == 'POST':
        todo.title = request.form['title']
        todo.des = request.form['desc']

```

```

        db.session.commit()
        return redirect("/")
    return render_template('update.html', todo=todo)

# Run the Flask app
if __name__ == "__main__":
    with app.app_context():
        db.create_all() # Ensure the database is created in the current
        directory
    app.run(debug=True)

```

5. Install databases like MySQL, MariaDB, etc.

## Usage

### Running the Application

Run the following command:

```
python app.py
```

### Output

Navbar MYTODO Link Dropdown Disabled Search Search

Todo title  
Enter your title

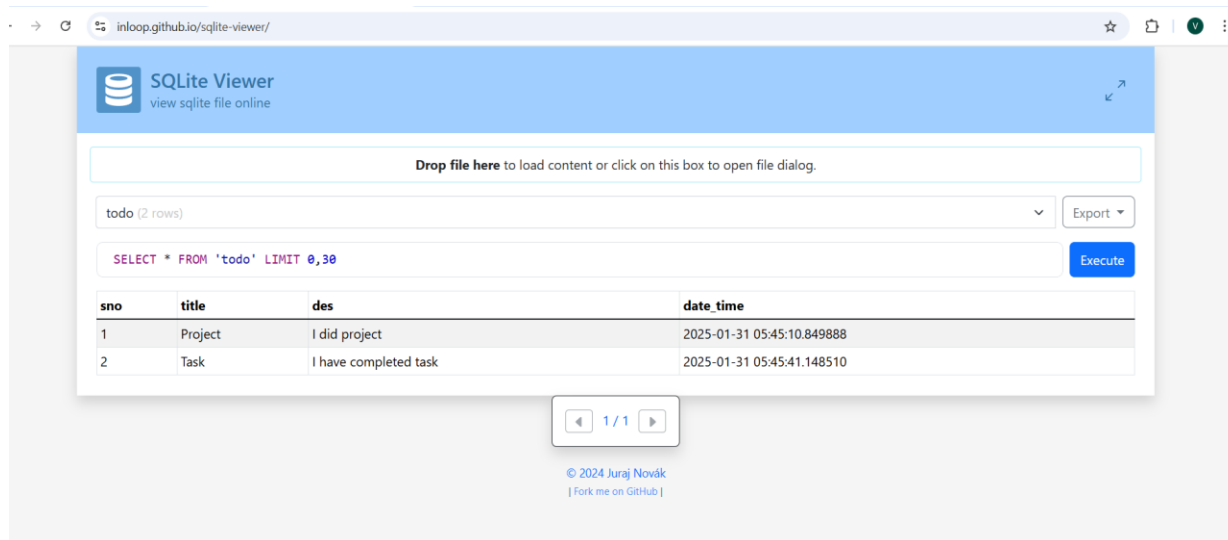
Description  
Enter your description

Submit

sno	title	desc	Time
1	Task	I have completed task	Delete Update
2	Project	I did project	Delete Update

After submitting the information, your data will be saved in the database.





If you want to delete or update your record, use the delete or update button.

Navbar

MYTODO Link Dropdown Disabled Search

Update Todo

Todo Title

Task

Description

I have completed task

Update Cancel

## API Endpoints (if using Flask/Django)

- **GET /tasks** - Retrieve all tasks
- **POST /tasks** - Add a new task
- **PUT /tasks/** - Update a task
- **DELETE /tasks/** - Delete a task

## Command-Line Interface (CLI)

If the app is CLI-based, use the following commands:

- `python app.py add "Task Title" "Task Description"`
- `python app.py list`
- `python app.py complete <task_id>`

- `python app.py delete <task_id>`

## Database Configuration

If using SQLite, the database file (`todo.db`) is automatically created. For other databases, update the configuration in `config.py`.

## Deployment

- **Local Deployment:** Run the application using `flask run` or `python app.py`.
- **Docker Deployment:** `docker build -t todo-app .`  
`docker run -p 5000:5000 todo-app`

