https://github.com/Nomi81/BanoQabil-2.0-Python-Course_Final_Project

BANO QABIL 2.0
FINAL PROJECT

“COURIER TRACKING
SYSTEM”

Contributors: Bilal Shakil & Noman Abbasi

Overview:

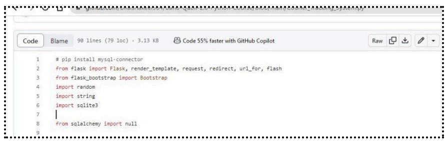
This Flask Based application serves as a basic Courier Tracking System allowing users to create shipments, track existing ones, update shipment details, and delete shipments. It utilizes SQLite for database management and integrates with Bootstrap for styling.

Dependencies:

Flask: Python web framework for building web applications.

Flask-Bootstrap: Integration of Bootstrap with Flask for easy UI development.

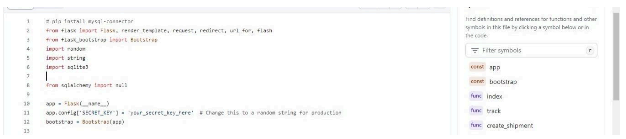
SQLite3: Database management system used for storing shipment data


Code
Blame
90 lines (79 loc) · 3.13 KB
Code 55% faster with GitHub Copilot
Raw

```
# pip install mysql-connector
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_bootstrap import Bootstrap
import random
import string
import sqlite3

from sqlalchemy import null
```

Coding Structure:

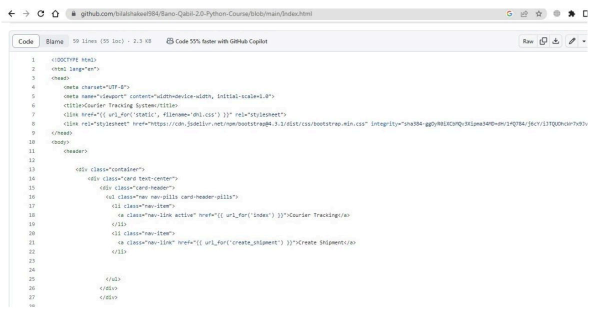
- app.py: Main Flask application file containing route definitions and database operations.



```python
# pip install mysql-connector
from flask import Flask, render_template, request, redirect, url_for, flash
from flask_bootstrap import Bootstrap
import random
import string
import sqlite3

from sqlalchemy import null

app = Flask(__name__)
app.config['SECRET_KEY'] = 'your_secret_key_here'  # Change this to a random string for production
bootstrap = Bootstrap(app)
```

Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.
Filter symbols
const app
const bootstrap
func index
func track
func create_shipment

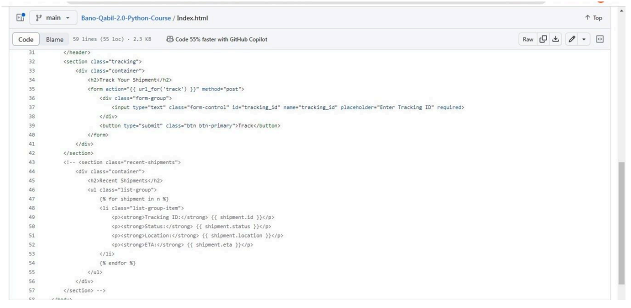
- index.html: Template for the homepage displaying the Courier Tracking system and a form to track shipments.



github.com/bilalshakeel984/Bano-Qabil-2.0-Python-Course/blob/main/Index.html

Code
Blame
59 lines (55 loc) · 2.3 KB
Code 55% faster with GitHub Copilot
Raw

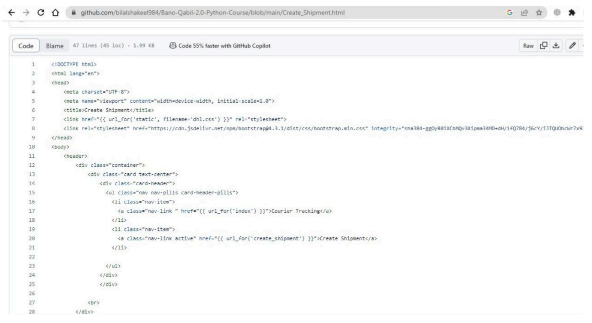
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Courier Tracking System</title>
    <link href="{{ url_for('static', filename='dhl.css') }}" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9Jv
</head>
<body>
    <header>

        <div class="container">
            <div class="card text-center">
                <div class="card-header">
                  <ul class="nav nav-pills card-header-pills">
                    <li class="nav-item">
                      <a class="nav-link active" href="{{ url_for('index') }}">Courier Tracking</a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="{{ url_for('create_shipment') }}">Create Shipment</a>
                    </li>


                  </ul>
                </div>
                </div>
```



main
Bano-Qabil-2.0-Python-Course / Index.html
Top
Code
Blame
59 lines (55 loc) · 2.3 KB
Code 55% faster with GitHub Copilot
Raw

```html
    </header>
    <section class="tracking">
        <div class="container">
            <h2>Track Your Shipment</h2>
            <form action="{{ url_for('track') }}" method="post">
                <div class="form-group">
                    <input type="text" class="form-control" id="tracking_id" name="tracking_id" placeholder="Enter Tracking ID" required>
                </div>
                <button type="submit" class="btn btn-primary">Track</button>
            </form>
        </div>
    </section>
    <!-- <section class="recent-shipments">
        <div class="container">
            <h2>Recent Shipments</h2>
            <ul class="list-group">
                {% for shipment in n %}
                <li class="list-group-item">
                    <p><strong>Tracking ID:</strong> {{ shipment.id }}</p>
                    <p><strong>Status:</strong> {{ shipment.status }}</p>
                    <p><strong>Location:</strong> {{ shipment.location }}</p>
                    <p><strong>ETA:</strong> {{ shipment.eta }}</p>
                </li>
                {% endfor %}
            </ul>
        </div>
    </section> -->
```

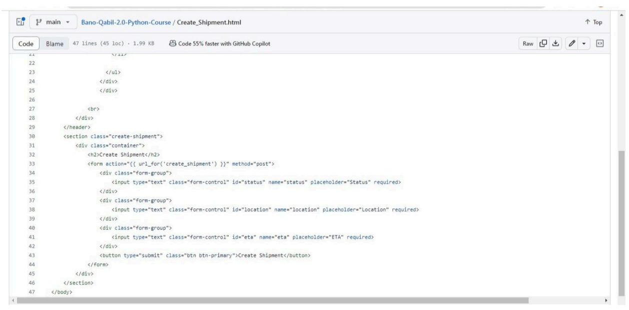
- create_shipment.html: Template for creating a new shipment with a form.



github.com/bilalshakeel984/Bano-Qabil-2.0-Python-Course/blob/main/Create_Shipment.html

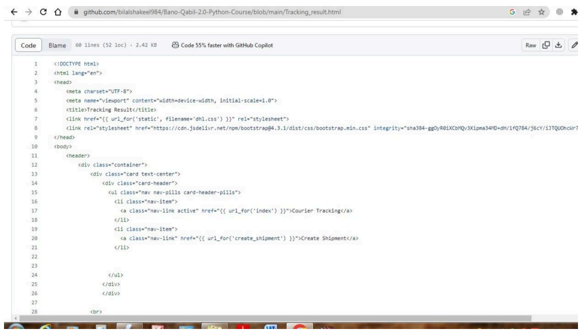
Code
Blame
47 lines (45 loc) · 1.99 KB
Code 55% faster with GitHub Copilot
Raw

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create Shipment</title>
    <link href="{{ url_for('static', filename='dhl.css') }}" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9
</head>
<body>
    <header>
        <div class="container">
            <div class="card text-center">
                <div class="card-header">
                  <ul class="nav nav-pills card-header-pills">
                    <li class="nav-item">
                      <a class="nav-link " href="{{ url_for('index') }}">Courier Tracking</a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link active" href="{{ url_for('create_shipment') }}">Create Shipment</a>
                    </li>

                  </ul>
                </div>
                </div>

            <br>
        </div>
```


main
Bano-Qabil-2.0-Python-Course / Create_Shipment.html
Top
Code
Blame
47 lines (45 loc) · 1.99 KB
Code 55% faster with GitHub Copilot
Raw

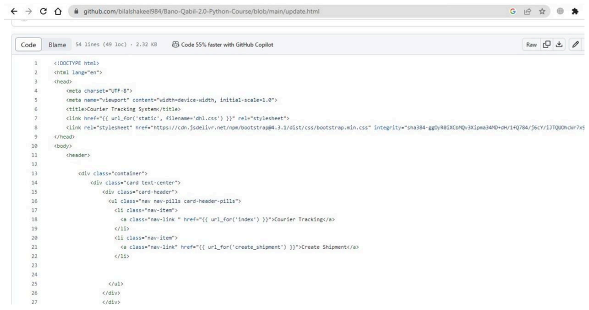
```html

                </ul>
            </div>
            </div>

          <br>
        </div>
    </header>
    <section class="create-shipment">
        <div class="container">
            <h2>Create Shipment</h2>
            <form action="{{ url_for('create_shipment') }}" method="post">
                <div class="form-group">
                    <input type="text" class="form-control" id="status" name="status" placeholder="Status" required>
                </div>
                <div class="form-group">
                    <input type="text" class="form-control" id="location" name="location" placeholder="Location" required>
                </div>
                <div class="form-group">
                    <input type="text" class="form-control" id="eta" name="eta" placeholder="ETA" required>
                </div>
                <button type="submit" class="btn btn-primary">Create Shipment</button>
            </form>
        </div>
    </section>
</body>
```

- tracking_result.html: Template to display the details of a tracked shipment and options to update or delete it.


github.com/bilalshakeel984/Bano-Qabil-2.0-Python-Course/blob/main/Tracking_result.html
Code
Blame
60 lines (52 loc) · 2.42 KB
Code 55% faster with GitHub Copilot
Raw

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tracking Result</title>
    <link href="{{ url_for('static', filename='dhl.css') }}" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7
</head>
<body>
    <header>
        <div class="container">
            <div class="card text-center">
                <div class="card-header">
                  <ul class="nav nav-pills card-header-pills">
                    <li class="nav-item">
                      <a class="nav-link active" href="{{ url_for('index') }}">Courier Tracking</a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="{{ url_for('create_shipment') }}">Create Shipment</a>
                    </li>


                  </ul>
                </div>
                </div>

            <br>
```

- update.html: Template for updating shipment details with a form.

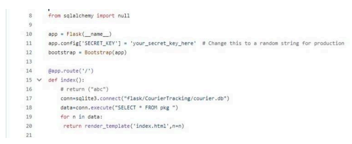

github.com/bilalshakeel984/Bano-Qabil-2.0-Python-Course/blob/main/update.html
Code
Blame
54 lines (49 loc) · 2.32 KB
Code 55% faster with GitHub Copilot
Raw

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Courier Tracking System</title>
    <link href="{{ url_for('static', filename='dhl.css') }}" rel="stylesheet">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9
</head>
<body>
    <header>

        <div class="container">
            <div class="card text-center">
                <div class="card-header">
                  <ul class="nav nav-pills card-header-pills">
                    <li class="nav-item">
                      <a class="nav-link " href="{{ url_for('index') }}">Courier Tracking</a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="{{ url_for('create_shipment') }}">Create Shipment</a>
                    </li>


                  </ul>
                </div>
                </div>
```

- dhl.css: CSS file for styling.

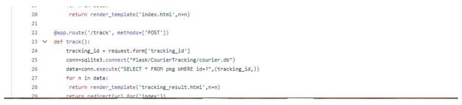
Functionalities:

1. Home Page (`/`):
    - Displays the Courier Tracking system.
    - Provides a form to track shipments.



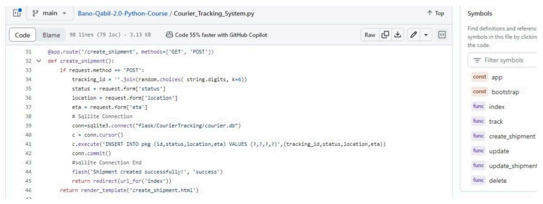
```python
from sqlalchemy import null

app = Flask(__name__)
app.config['SECRET_KEY'] = 'your_secret_key_here'  # Change this to a random string for production
bootstrap = Bootstrap(app)

@app.route('/')
def index():
    # return ("abc")
    conn=sqlite3.connect("flask/CourierTracking/courier.db")
    data=conn.execute("SELECT * FROM pkg ")
    for n in data:
     return render_template('index.html',n=n)

```

2. Track Shipment (`/track`):
    - Retrieves shipment details based on the provided tracking ID.
    - Renders the tracking result page with shipment details.



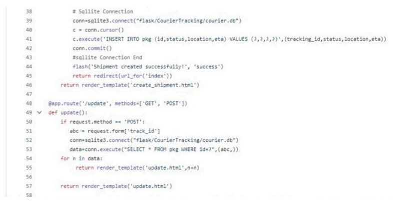
```python
     return render_template('index.html',n=n)

@app.route('/track', methods=['POST'])
def track():
    tracking_id = request.form['tracking_id']
    conn=sqlite3.connect("flask/CourierTracking/courier.db")
    data=conn.execute("SELECT * FROM pkg WHERE id=?",(tracking_id,))
    for n in data:
     return render_template('tracking_result.html',n=n)
    return redirect(url_for('index'))
```

3. Create Shipment (`/create_shipment`):
    - Allows users to create new shipments.
    - Generates a random tracking ID.
    - Inserts shipment details into the database.


main
Bano-Qabil-2.0-Python-Course / Courier_Tracking_System.py
Top
Code
Blame
90 lines (79 loc) · 3.13 KB
Code 55% faster with GitHub Copilot
Raw
Symbols
Find definitions and referenc
symbols in this file by clickin
the code.
Filter symbols
const app
const bootstrap
func index
func track
func create_shipment
func update
func update_shipment
func delete

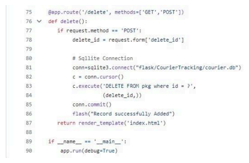
```python
@app.route('/create_shipment', methods=['GET', 'POST'])
def create_shipment():
    if request.method == 'POST':
        tracking_id = ''.join(random.choices( string.digits, k=6))
        status = request.form['status']
        location = request.form['location']
        eta = request.form['eta']
        # Sqllite Connection
        conn=sqlite3.connect("flask/CourierTracking/courier.db")
        c = conn.cursor()
        c.execute('INSERT INTO pkg (id,status,location,eta) VALUES (?,?,?,?)',(tracking_id,status,location,eta))
        conn.commit()
        #sqllite Connection End
        flash('Shipment created successfully!', 'success')
        return redirect(url_for('index'))
    return render_template('create_shipment.html')
```

4. Update Shipment (`/update` and `/update_shipment`):
    - Allows users to update shipment details.
    - Retrieves shipment details based on the provided tracking ID.
    - Renders the update page with pre-filled shipment details.
    - Updates shipment details in the database upon submission.



```python
            # Sqllite Connection
            conn=sqlite3.connect("flask/CourierTracking/courier.db")
            c = conn.cursor()
            c.execute('INSERT INTO pkg (id,status,location,eta) VALUES (?,?,?,?)',(tracking_id,status,location,eta))
            conn.commit()
            #sqllite Connection End
            flash('Shipment created successfully!', 'success')
            return redirect(url_for('index'))
        return render_template('create_shipment.html')

    @app.route('/update', methods=['GET', 'POST'])
    def update():
        if request.method == 'POST':
           abc = request.form['track_id']
           conn=sqlite3.connect("flask/CourierTracking/courier.db")
           data=conn.execute("SELECT * FROM pkg WHERE id=?",(abc,))
        for n in data:
             return render_template('update.html',n=n)

        return render_template('update.html')
```

5. Delete Shipment (`/delete`):
    - Deletes a shipment from the database based on the provided tracking ID.



```python
@app.route('/delete', methods=['GET','POST'])
def delete():
   if request.method == 'POST':
        delete_id = request.form['delete_id']

        # Sqllite Connection
        conn=sqlite3.connect("flask/CourierTracking/courier.db")
        c = conn.cursor()
        c.execute('DELETE FROM pkg where id = ?',
                  (delete_id,))
        conn.commit()
        flash("Record successfully Added")
   return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Database Structure

- Table Name: `pkg`
- Columns: `id` (tracking ID), `status`, `location`, `eta` (estimated time of arrival)