

1.Open Anaconda Navigator

2.Create Environments

3.Open Terminal

4.Path

5.Run Command: flask --app app run --debug

Anaconda Navigator

- Is a Graphical User Interface(GUI) that's part of the Anaconda distribution for python.
 - It allows users to manage conda packages, environments,& channels.
-

Flask Framework :

- Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. Lightweight.
 - Flask is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.
 - It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file.
 - It used for backend development
 - Flask handles the back-end with routes and data handling.
-

Creating a Web Application

- 1.Initializing the flask app
 - 2.Creating a database
 - 3.Defining Routes
 - 4.Creating Templates(html)
 - 5.Running the Application
-

Routes -

- routes system for mapping URL's to Controllers/Actions and generating URL's
 - Flask decorator that you have probably used is the @app.route('/') for defining routes. When displaying the output to a browser, this decorator converts a function into a route that can be accessed by the browser without having to explicitly invoke the function in the program.
 - Routes in Flask are defined using the @app.route() decorator. This decorator takes the URL path as its argument and associates it with a specific view function.
-

SQLite

-**SQLite** is a self-contained, serverless, and zero-configuration database engine. It is perfect for small to medium-sized web applications and mobile apps, as it doesn't require a separate server or complex setup.

-**To use SQLite in a Python Flask project, you can follow these steps:**

1.**To install required modules** : pip install Flask sqlite3

2.**Create Database connection** :

```
import sqlite3

conn = sqlite3.connect('database.db')
```

3.**Create a cursor**: A cursor is an object that you can use to execute SQL statements. You can create a cursor by calling the cursor() method on the database object.

```
cur = conn.cursor()
```

4.**Execute SQL statements**: You can execute SQL statements using the cursor object. For example, to create a table, you would use the following SQL statement:

```
cur.execute("""CREATE TABLE users (id INTEGER PRIMARY KEY, name TEXT)""")
```

5.**Commit changes**: Once you have made changes to the database, you need to commit them. You can do this by calling the commit() method on the database object.

6.**Close the database connection**.: When you are finished using the database, you need to close the connection. You can do this by calling the close() method on the database object.

Project

Code:

execution

1.app.py

2.routing work

step

1.front end - html,css js

2.backend - py(flask)(admin user & db model(table str creation), page routing, prediction logic,)

3.db - SQLite(table create logic flask madhe use)

4.getting data set(raw data set)()

5.data set cleaning process

6.train model using random forest alg.(model.pkl generate)

7.clean data set model used in backend to prediction. (backend : model.pkl file)
